# A FRAMEWORK FOR CONDUCTING MECHANISTIC BASED RELIABILITY ASSESSMENTS OF COMPONENTS OPERATING IN COMPLEX SYSTEMS

A Thesis
Presented to
The Academic Faculty

by

**Jon Michael Wallace**

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
November 2003

# A FRAMEWORK FOR CONDUCTING MECHANISTIC BASED RELIABILITY ASSESSMENTS OF COMPONENTS OPERATING IN COMPLEX SYSTEMS

Approved by:

| | |
|---|---|
| Dr. Dimitri Mavris, Advisor | Dr. James Craig |
| Dr. Daniel Schrage | Dr. Ajay Misra |
| Dr. Richard Neu | |

Date Approved: November 24, 2003

*To my family;*

*my parents,*

*my brother,*

*my wife,*

*and my children.*

# ACKNOWLEDGEMENTS

I would like to express sincere appreciation to my thesis advisor, Prof. Dimitri Mavris, for his tremendous guidance and support throughout my research. Profs. Daniel Schrage and Richard Neu are to be thanked for their involvement during this study. I wish to thank Prof. James Craig and Dr. Ajay Misra for their support as thesis reading committee members. Drs. Zhimin Liu and Vitali Volovoi and Mrs. Rhenua Wang are to be thanked for the many fruitful and enlightening discussions related to my thesis study. I also wish to thank my fellow peers and friends, Dr. Zhimin Liu, Brian German, Andrew Frits, Rob McDonald, Taewoo Nam, and Peter De Baets, who have unselfishly assisted with the final editorial review of my thesis. Lastly, my parents deserve thanks for enduring many years of youthful mischief and especially for providing tremendous assistance and support along the way; without which my accomplishments would have been difficult if not impossible to attain. Thank you all.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# SUMMARY

Reliability prediction of components operating in complex systems has historically been conducted in a statistically isolated manner. Current physics-based, i.e. mechanistic, component reliability approaches focus more on component-specific attributes and mathematical algorithms and not enough on the influence of the system. The result is that significant error can be introduced into the component reliability assessment process.

The objective of this study is the development of a framework that infuses the needs and influence of the system into the process of conducting mechanistic-based component reliability assessments. The formulated framework consists of six primary steps. The first three steps, identification, decomposition, and synthesis, are primarily qualitative in nature and employ system reliability and safety engineering principles to construct an appropriate starting point for the component reliability assessment.

The following two steps are the most unique. They involve a step to efficiently characterize and quantify the system-driven local parameter space and a subsequent step using this information to guide the reduction of the component parameter space. The local statistical space quantification step is accomplished using two proposed multivariate probability models: Multi-Response First Order Second Moment and Taylor-Based Inverse Transformation. Where existing joint probability models require preliminary distribution and correlation information of the responses, these models combine statistical information of the input parameters with an efficient sampling of the response analyses to produce the multi-response joint probability distribution.

Parameter space reduction is accomplished using Approximate Canonical Correlation Analysis (ACCA) employed as a multi-response screening technique. The novelty

of this approach is that each individual local parameter and even subsets of parameters representing entire contributing analyses can now be rank ordered with respect to their contribution to not just one response, but the entire vector of component responses simultaneously.

The final step of the framework is the actual probabilistic assessment of the component. Although the same multivariate probability tools employed in the characterization step can be used for the component probability assessment, variations of this final step are given to allow for the utilization of existing probabilistic methods such as response surface Monte Carlo and Fast Probability Integration.

The overall framework developed in this study is implemented to assess the finite-element based reliability prediction of a gas turbine airfoil involving several failure responses. Results of this implementation are compared to results generated using the conventional 'isolated' approach as well as a validation approach conducted through large sample Monte Carlo simulations. The framework resulted in a considerable improvement to the accuracy of the part reliability assessment and an improved understanding of the component failure behavior. Considerable statistical complexity in the form of joint non-normal behavior was found and accounted for using the framework. Future applications of the framework elements are discussed.

# CHAPTER I

# INTRODUCTION

Since the first documented failure experiments by Wöhler in 1860 [98], researchers have acknowledged that considerable variation can be expected for many types of failure mechanisms. In practice, engineers have dealt with this unfavorable variation by introducing large margins in critical failure parameter predictions resulting in increased weight and cost. However, this safety factor approach is severely limiting when applied to complex systems such as aircraft. These systems are subject to increasingly more severe constraints on performance and utilization that challenge the safety factor approach. For instance, aeronautical vehicles have a strong weight reduction requirement. Further, economic factors have resulted in an increased impetus for reducing the production and even operational cost of the systems. Noteworthy advances in structural materials, component design, and related production processes have increased the capability of engineering systems considerably. However, unanticipated failures, some even catastrophic, have continued to plague such systems. Several historically significant reliability related events include:

*Reliability*: Historical Perspective

- Early 1930's with electric power generation and aircraft engine studies
- Germans applied basic concepts to V1 and V2 rockets (WWII)
- Post WWII era saw a shift towards more complex systems with more stringent development schedule and cost constraints
- 1st electronic reliability study (Cornell and Aeronautical radio in 1947)

- In 1950, US DOD established first committee on reliability
- In 1951 Weibull published the famous Weibull statistical function
- De Havilland Comet accidents of the early 1950's
- 1st National symposium on reliability and quality control in 1954 just before IEEE launched the Reliability and Quality Control Society
- In 1962, Air Force Institute of Technology began the 1st Masters degree program in system reliability engineering
- Point Pleasant bridge disaster (1967)
- Early 1970's increase in construction of large nuclear and fossil fuel power plants, large commercial airliners (public safety concern)
- Space shuttle Challenger accident and Chernobyl reactor failure (1986)
- TWA Flight 800 disaster (1996)
- Alaskan airlines Flight 261 (2000)
- Recent space shuttle Columbia failure (2003)

The application of probability and statistics to failure analysis, a necessary ingredient in reliability, is considered to be the leading candidate for understanding and preventing unwanted failures from occurring. Even though this field is considered to be mature; there are several areas where researchers are focused on continuing its development. An area that has potential for improvement, and the subject of this thesis, is that of probabilistic failure analyses, or reliability, of components operating in complex systems.

## *1.1 Motivation*

Reliability is defined as *the probability that an item will survive a specified amount of time under specified conditions.* Reliability has become increasingly popular, particularly in the electronics, aviation, nuclear, and petroleum industries. The reason for an increased focus on reliability can be attributed to continued public demand to improve the safe operation of such systems as well as economic and environmental factors that demand that such systems perform efficiently.

### 1.1.1 Reliability and Public Safety

One area of public interest that can benefit from reliability analyses is that of the National Aerospace System. With more of the population travelling every year, the demand placed on the aerospace system is expected to increase exponentially. According to a 1997 White House Presidential Commission report on Aviation Safety and Security of the National Aerospace System [44], almost 300 million people flew in the year 1980. The number of people flying in the year 2007 is expected to triple that of 1980. As of 1996, there were only 0.3 fatal accidents per million flight hours, which is comforting. However, with the ever increasing demand for air travel, the number of accidents is expected to increase substantially even as the fatal accident rate remains stable. For instance, within the 1997 White House report report Boeing is predicts that if the current accident rate is not reduced, there will be a major accident somewhere in the world every week due to the number of future flights that are projected! Since this commission disclosed its findings, the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration have undertaken the presidential directive to improve aviation safety [5]. Specifically, the 2001 safety strategy created by these two entities has a primary goal of reducing the fatal accident rate by 80% of 1997 levels before the year 2007. One way of doing so is to improve both the process and tools for designing reliable systems.

### 1.1.2 Economics of Failure

The economy is another force that adds to the need for reliability assessments. Failure of engineering systems has historically contributed to significant loss in revenue. Reed reported in 1983 that the annual cost of fracture of materials in the U.S. resulted in a loss of $119 Billion which was 4% of the Gross National Product [75]. To illustrate the importance of aerospace systems, the commercial aerospace sector was reported within the White House Aviation Safety Commission to be generating over $300 Billion annually by the year 1996. And, while developers of complex systems have historically ensured safe designs, performance rather than reliability of these systems has been the primary focus. However, a paradigm shift is taking place. Deregulation of several industries, corporate cost-cutting across most sectors, and other factors have lead to increased competition amongst companies that develop and produce such systems. Due to strong price competition, profits from the sale of these products have steadily declined. This is especially true in the aircraft and aircraft engine industries where it is not uncommon for a company to sell an engine or aircraft near or even below cost. With the cost of these products held to a minimum, companies have looked to other areas to remain competitive. One such area is that of reliability. Designing a product that can not only perform competitively but also can do so reliably has become a major focus during the development of these products. However, many of these products are quite complex and even traditional physical analyses can be quite demanding let alone probabilistic studies required for reliability assessments.

## *1.2 System Reliability*

Before the concepts of mechanistic-based reliability of components are given, a cursory account of a complex system is to be given. A system is broadly defined as "the entire combination of hardware, information, and people necessary to accomplish some specified mission." [26]. Blanchard and Fabrycky [15] provide a more refined

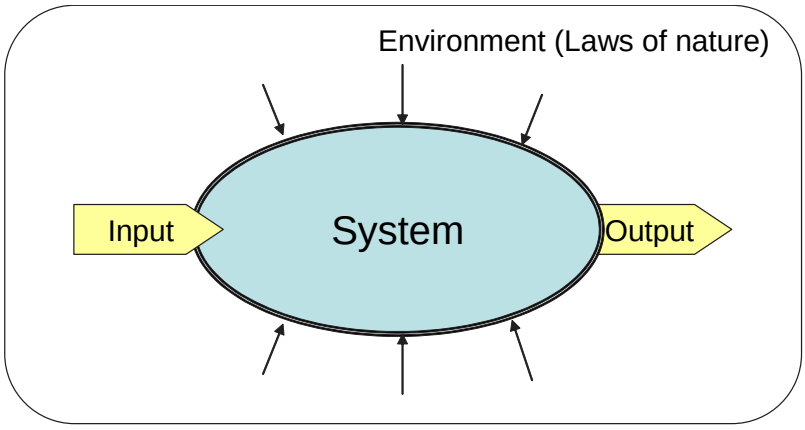


**Figure 1:** System and Environment Interaction.

definition stated as "the total system, at whatever level in the hierarchy, consists of all components, attributes, and relationships needed to accomplish an objective. Each system has an objective, providing a purpose for which all system components, attributes, and relationships have been organized". As shown in Figure 1, constraints placed on the system limit its operation and define the boundary within which it operates. Likewise, the system places boundaries and constraints on its subsystems and therefore its components as well.

Reliability assessment of complex systems is a very challenging task, requiring interaction between many traditional disciplines in addition to the use of probability and statistics. Further, component-level reliability analysis usually occurs after much of the top-level system engineering and design has already occurred. As a result, important component information, which system level analyses are based on, is not utilized. Therefore, component reliability should be considered earlier in the product development process.

Such a vision of concurrent development between system safety/reliability and the product life cycle is depicted in Figure 2. While running the gamut of product development activities, safety assessment activities such as Fault Hazard Assessment

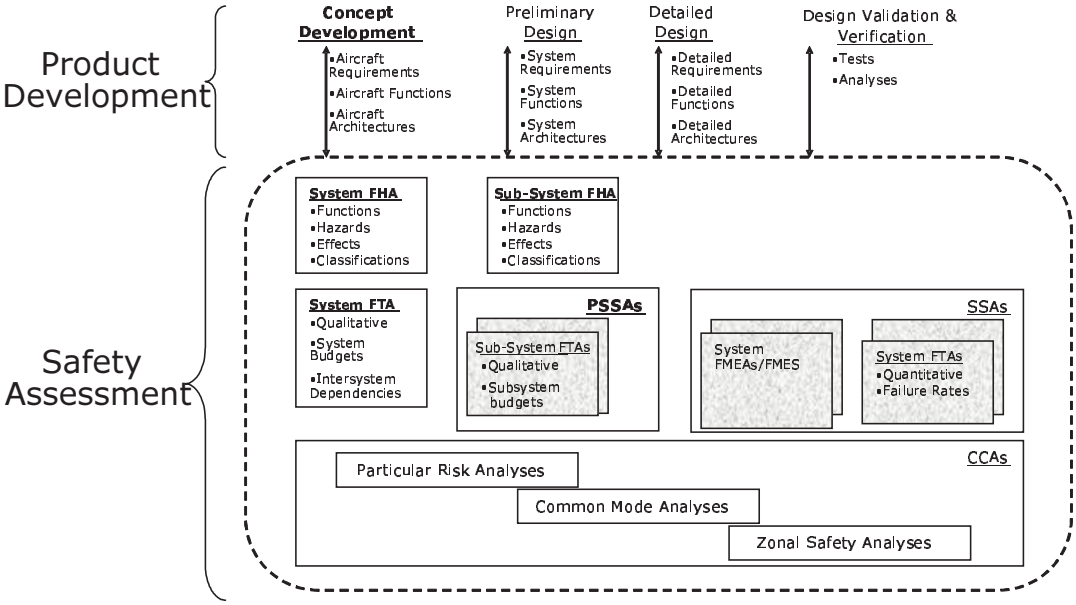


**Figure 2:** Concurrent Safety/Reliability Assessment and Product Development Cycle Processes [3].

(FHA) and Fault Tree Assessments (FTA) at the top-level down are conducted initially in a qualitative and eventually in a quantitative manner. These activities are infused early on at the system level and continued later during Preliminary System Safety Assessments (PSSA) of the sub-systems. Numerous common cause assessments (CCAs) between the various subsystems are recommended throughout this process. The approach shown in Figure 2 provides the basis for scheduling reliability allocations to each sub-system to meet the system safety/reliability goal. These individual sub-system allocations must eventually be validated and acceptable assurance levels reached for meeting the system reliability goals. It is this phase where component-level reliability assessments are of great importance as they can be used to provide assurance that the top-level system reliability and safety goals are met.

## *1.3 Component Reliability*

Assessing the reliability of a component requires the involvement of four major areas; loading, geometry, materials, and probability, as depicted in Figure 3. The reliability

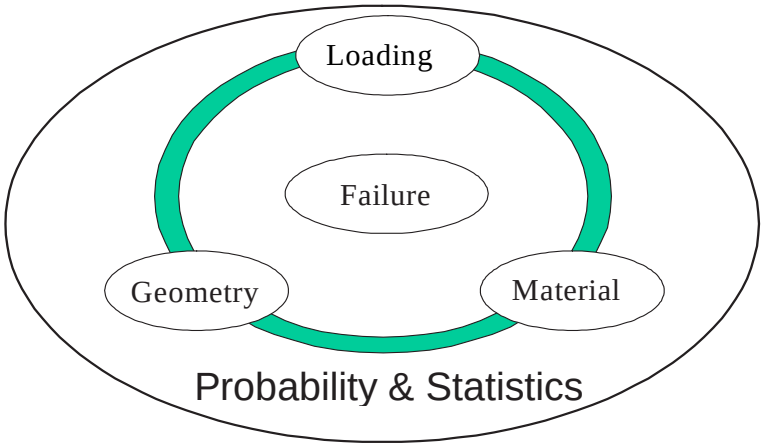


**Figure 3:** Facets of Component-Level Reliability Assessment

analysis of a gas turbine blade, such as the one depicted in Figure 4, is used here to demonstrate the interrelationship between the areas shown in Figure 3. The loading area consists of defining the initial and boundary aero-thermo-mechanical conditions. This is the most complicated and resource-intensive area due to the fact that these conditions are a function of a multitude of upstream physical phenomena. This area includes determining the thermodynamic behavior as well as the local flow characteristics such as aerodynamic and aerothermal properties. Further complicating the area of load determination is the fact that these loads are highly dependent on the operating and environmental conditions of the system. As will be discussed later, the influence of upstream physical behavior on component reliability is difficult to discern and is frequently neglected.

The geometry of the component is also an important area of consideration when conducting failure analyses. Typically, this is the area under the most control of the designer. But, depending on the intricacy of the geometric features, severe nonlinearity can be introduced into the simulation process. For instance, features such as bolt holes, fillets, and part transition regions can greatly affect the local stress distribution of the part under loading. Variations in these key features can have a significant effect on the failure and probabilistic analysis of the component.

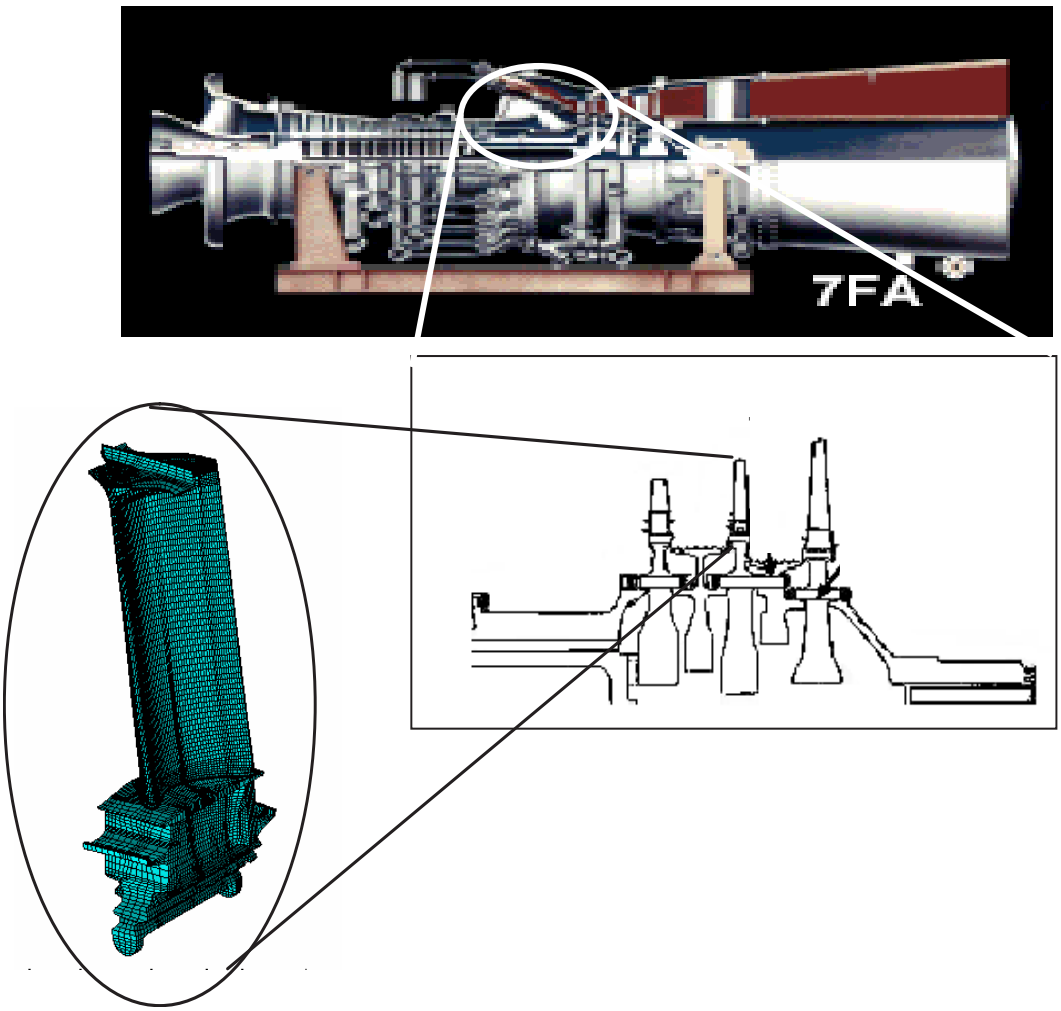


**Figure 4:** Representative Complex System Component: Gas Turbine Blade.

Material properties are just as important in the reliability assessment of components. It is the material properties that define how the material responds to the loads applied to the component. For a given condition, the loading and geometric properties define the internal force intensity and other mechanical states of the material. The material properties then are used to describe how the material will react to such force intensity. There can be strong interaction between the loading, geometry, and material properties areas in describing the current thermo-mechanical state of a component. These interactions are accounted for during the failure analysis.

The failure area of physics-based reliability is concerned with integrating the triad of geometry, loading, and material properties to determine when, where, as well as how the component might fail. This area has been the beneficiary of considerable research during the past century particularly for engineering materials. Through this research as well as actual field failure experience, numerous failure mechanisms have been identified, studied, and modelled. These failure mechanisms can be organized into five main categories; deformation, fracture, corrosion, wear, and erosion [88].

Deformation failure occurs when the shape or size of the component is altered to a point where the part can no longer function properly. Fracture occurs when a crack initiates within the component and propagates to a critical size causing the complete separation of the component. Researchers in the area of mechanics of solids have contributed significantly to the understanding and modeling of deformation and fracture. Corrosion is the degradation or even loss of material due to chemical processes while wear is a situation where the surface of two or more components is altered by continual abrasive contact between them. Finally, erosion can be considered a special case of wear where the abrasive condition is caused by continual contact between a gas containing abrasive particulates and a solid surface.

The last area depicted in Figure 3 is that of probability and statistics. Techniques within this area must be applied to some or all of the four physics-based areas in

order to assess the failure probability, and of course its complement, reliability, of the component in question. It is this area and its interaction with the four physics-based areas that will be investigated within the proposed research.

### 1.3.1 Limitations of Component Reliability

While many probabilistic techniques have evolved to permit numerically accurate probabilistic assessments based on the underlying physics, they do so in a rather isolated manner with respect to the complex interaction that exists between a component and the system it supports. Considering the component in an isolated manner is an advantageous simplification to the highly interdisciplinary analysis required of components of complex systems. Obvious time and monetary savings can be realized by pursuing such an 'isolated' approach. This is because an integrated approach, using the current state of technology, can be prohibitive to evaluate probabilistically and sometimes even deterministically. However, such an isolated approach has been shown to be severely limiting, statistically, in that it can potentially fail to consider critical statistical phenomena due to common yet over-simplifying assumptions [93].

Some of these unfettered assumptions include normality and independence of the random input variables local to the component analysis. Non-normality and statistical dependence are two conditions that should be considered, especially for components of highly integrated, complex systems. Should either of these two conditions exist and go unaccounted for, considerable inaccuracy in the probabilistic analysis can be expected. Ideally, such local statistical phenomena would be accounted for by a highly integrated, interdisciplinary environment. Unfortunately, such a framework exists at best only in a partial sense in the realm of applied research and rarely in practice. In practical terms, there are tremendous organizational, computational, and resource-related barriers to constructing such an environment. Therefore, a refined method for efficiently characterizing the statistical phenomena of local, random variables of

components of complex systems is desired. Ideally, this method would enhance the conventional isolated approach and provide a suitable balance between statistical accuracy and computational as well as organizational efforts. The development of such a method is the objective of the proposed research.

## 1.4 Research Goals

The focus of this research is to develop a framework that integrates probability theory and physics-based models for improved probabilistic failure analysis of components operating in complex systems. A foreseeable accomplishment of such a framework, firstly, is the early characterization of the statistical behavior between the system and the component. Such a characterization should account for joint randomness of the component reliability parameters in addition to significant non-Gaussian behavior. Secondly, using this characterization a probabilistic framework is sought that can efficiently provide component reliability as a function of global and local variables. Lastly, the framework should be amenable to the system design process and therefore consistent with conventional multi-disciplinary design and optimization techniques.

## 1.5 Research Questions

In meeting the goals of the proposed research, there are several questions that are to be addressed. They include:

1. What are the limitations of currently accepted methods used in industrial probabilistic assessment activities?

2. What statistical behavior is expected for components of complex systems, such as a gas generator turbine blade?

3. What advances in probability and statistics theory have been made which show promise in overcoming such limitations?

4. Are new advances in probability and statistics theory required?

5. What would be an ideal framework of such advances and accepted methods?

# CHAPTER II

# BACKGROUND

The concept of reliability has been acknowledged for quite some time but didn't appear in a technical sense until the early 1930's as a means of providing design guidance for selecting the number of airplane engines for propeller driven aircraft. The field has since seen spurts of growth in various industries such as aerospace, electronics, bio-technology, and even actuarial science. Recent emphasis on this field has increased as the relevant mathematical basis becomes more generally known and the benefits of predicting the life and functionality of critical items are becoming more apparent. Although considerable advances have been made within the reliability discipline, the methods and tools of reliability have not, until recently, been introduced to the realm of complex system design and analysis.

## *2.1 Component Failure Analyses*

The research proposed within this study focuses on the probabilistic failure analysis of a component as a function of system variation. A representative component, a turbine blade of an air-breathing gas turbine engine, has been selected to provide the context from which the study will be conducted. This component is representative in that it experiences several modes of failure, has historically been the focus of much research effort, and is a critical component within a gas turbine generator. An account of the literature related to deterministic failure analyses of a turbine blade is now provided.

Turbine blade analysis requires the involvement of many disciplines and the use of advanced computational and experimental techniques such as sophisticated aerodynamic, thermal, and mechanical experiments and especially multi-dimensional CFD

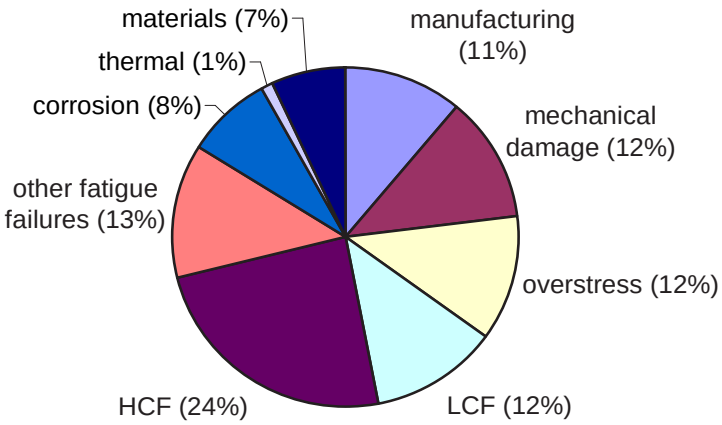


**Figure 5:** Failure Mechanisms in Aircraft Jet Engines [22].

and FEA models. There are a multitude of reasons why an airfoil would fail. For instance, complex multi-axial stress states exist due to pressure distribution behavior, buffeting or vibratory loadings, flutter, temperature gradients, local geometric non-linearity, and maneuver loads. The problem is compounded by the fact that the airfoil is exposed to a highly aggressive medium that promotes erosion, corrosion, and even creep of the airfoil. Cyclic loadings arising from these loads can cause microscopic damage to initiate and promulgate into cracks and eventual fracture of the airfoil.

A list of several failure modes and their relative contribution for jet aircraft engines was studied by Cowles [22]. Cowles's results are shown in Figure 5. Notice that high cycle, low cycle, and other fatigue modes accounted for almost half of the jet engine failures. The crutch of many turbine blade deterministic failure analyses is the tremendous infrastructural and interdisciplinary effort required particularly when considering system parameters. However, variation at the system level is a major contributor to the component level failure uncertainty.

### 2.1.1 Modeling Operating and Environmental Conditions

Only a handful of researchers have addressed the need for modelling system operating conditions when conducting component probabilistic failure assessments. Liu et al.

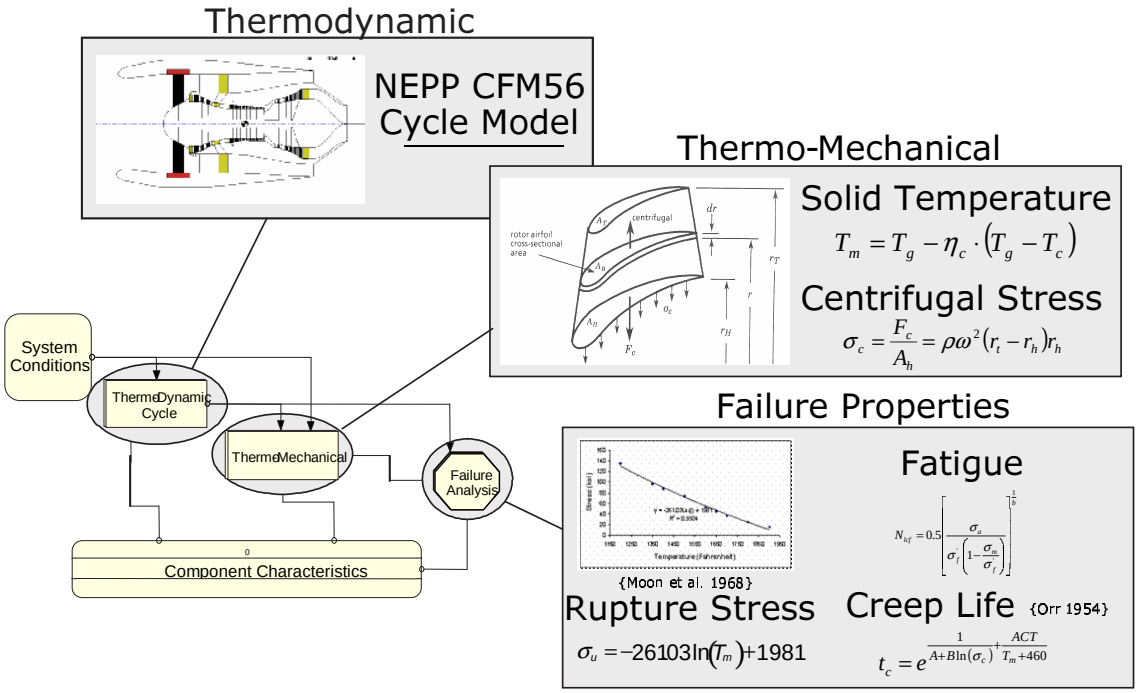


**Figure 6:** Multi-Disciplinary Failure Analysis Structure [93].

[60] demonstrated a means of modelling local component operating conditions such as gas path flow temperature and rotational speed of a turbine airfoil, which can be determined online during operation using sensorial data. Liu [59] subsequently extended this work to include a nondeterministic consideration of the local variables with deterministic local operating conditions as a way to provide real-time remaining life estimation given actual sensorial information of the local blade parameters. Top-level versus local operating conditions were modeled by Wallace and Mavris [93] using a relatively simple integrated turbine blade analysis environment, shown in Figure 6. The focus of this study revolved around estimating more accurate local variable statistical properties based on the driving top-level system conditions such as environmental and operating conditions using an integrated, multi-level/multi-disciplinary environment. Not only did the integrated approach enable system level parameters to be included in a component level probabilistic analysis, it also inherently modelled local variable non-normality and dependence. This was a significant

improvement over traditional isolated approaches that assumed independence of the local parameters. These parameter assumptions were shown to be invalid, introducing considerable error into the reliability analysis of a turbine blade. However, this particular study was very basic with respect to the physical analyses used and was conducted to further identify the problem or need for modelling complex statistical behavior driven by the system-component interaction.

### 2.1.2 Multidisciplinary Environments

There are several researchers that have begun to consider the integration of at least some of the areas shown in Figure 3 within a framework for a single, complex component. Environments, such as FIPER (Burton [20], Tappeta [89]), MOPED (Jeschke et al. [50]) and a gas turbine blade integrated analysis tool developed by Tinga et al. [90], have been created that couple multiple local analyses representative of some if not all of the areas shown in Figure 3. The analyses are deterministic and are considered a major improvement in computational capability required for providing adequate consideration of multiple disciplines necessary for component simulation. Meta-model[1] techniques, also used in FIPER, have become popular in creating more manageable mathematical realizations of these disciplines within a system. Several researchers have recently applied these approximation techniques to complex component analyses [93][94][51][64]. A multidisciplinary environment such as one of the many mentioned here would provide a much needed foundation for unveiling the complex statistical phenomena of the local component failure parameter space.

[1]Metamodel is a general term for a more manageable functional relationship of a much more complex and intractable one. An example is a response surface equation created from a more involved structural analysis.

## 2.2 Probability and Statistics Preliminaries

The proposed reliability research covers a broad range of topics within the area of probability and statistics. Therefore, a brief summary of relevant probabilistic and statistical theory is now given. The summary includes univariate and multivariate random variable probability theory.

Univariate probability has now become standard material within accredited undergraduate engineering curricula. The concept of the probability distribution and cumulative distribution functions for both continuous and discrete variables are common. Also, several univariate distributions have seen wide spread use across several fields within engineering and science (e.g. the normal, exponential, Weibull, and log-normal). Statistical theories that describe how to make inferences from sample sets of a general population are also common as a means of providing the data analysis tools necessary to complement probability theory. These theories are extremely important in that they provide a means of estimating parameters used in probabilistic analyses. Several textbooks have been written covering these basic concepts (see for example Devore [30]).

### 2.2.1 Multivariate Probability

When considering structural applications multivariate probability and statistics are usually required. The univariate probability density function for a continuous variable can be generalized for multiple random variables. For two random variables that are jointly distributed the probability that they take on values within a specified range is expressed as

$$P[(X,Y) \in A] = P(a \leqslant X \leqslant b, c \leqslant Y \leqslant d) = \int_a^b \int_c^d f(x,y)dydx \qquad (1)$$

where $A$ is the event probability space and $a$,$b$,$c$, and $d$ are the interval end points under consideration. One can further expand this equation to consider a multivariate

function of three or more jointly distributed random variables given as

$$P[(X_1, X_2, \ldots, X_n) \in A] = \int_A \ldots \int_A f(x_1, x_2, \ldots, x_n) dx_1 dx_2 \ldots dx_n \tag{2}$$

It is apparent that the exact solution of this integral could be difficult, if not impossible to find. If the variables are dependent, the evaluation of this integral is further complicated. The joint distribution between them is then required.

An example of a joint distribution function with a closed form solution is the bivariate normal function and is expressed as

$$\begin{aligned} f_{XY}(x,y) = & \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_{x,y}^2}} \\ & exp\left\{\frac{1}{2\rho_{x,y}^2-2}\left[\left(\frac{x-\mu_x}{\sigma_y}\right)^2 - 2\rho_{x,y}\left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right) + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]\right\} \end{aligned} \tag{3}$$

where $\sigma_x$ and $\sigma_y$ are the standard deviation of the $X$ and $Y$ random variables, $\mu_x$ and $\mu_y$ are the mean values of these two random variables, and $\rho_{x,y}$ is the correlation coefficient of $X$ and $Y$. This solution requires the knowledge of only these five parameters. The bi-variate normal probability density function is illustrated in Figure 7. Notice the added dimensionality of integrating over select ranges necessary for probabilistic calculations.

The correlation coefficient shown in equation (3) is a metric used to quantify linear, statistical dependence between two variables. To define the correlation coefficient one must first define covariance. The covariance of two random variables is the second moment about the mean and is given by

$$\begin{aligned} COV(X_i, X_j) &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}(x_i-\mu_{x_i})(x_j-\mu_{x_j})f(x_{x_i,x_j})dx_i dx_j \\ &= E[(X_i-\mu_{x_i})(X_j-\mu_{x_j})] = E[X_iX_j] - \mu_{x_i}\mu_{x_j} \end{aligned} \tag{4}$$

$$\text{where } E[X_iX_j] = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} x_i x_j f(x_{x_i,x_j})dx_i dx_j \tag{5}$$

The covariance between any two variables can have awkward units and vastly different magnitudes. A more useful term is the correlation coefficient, which normalizes the

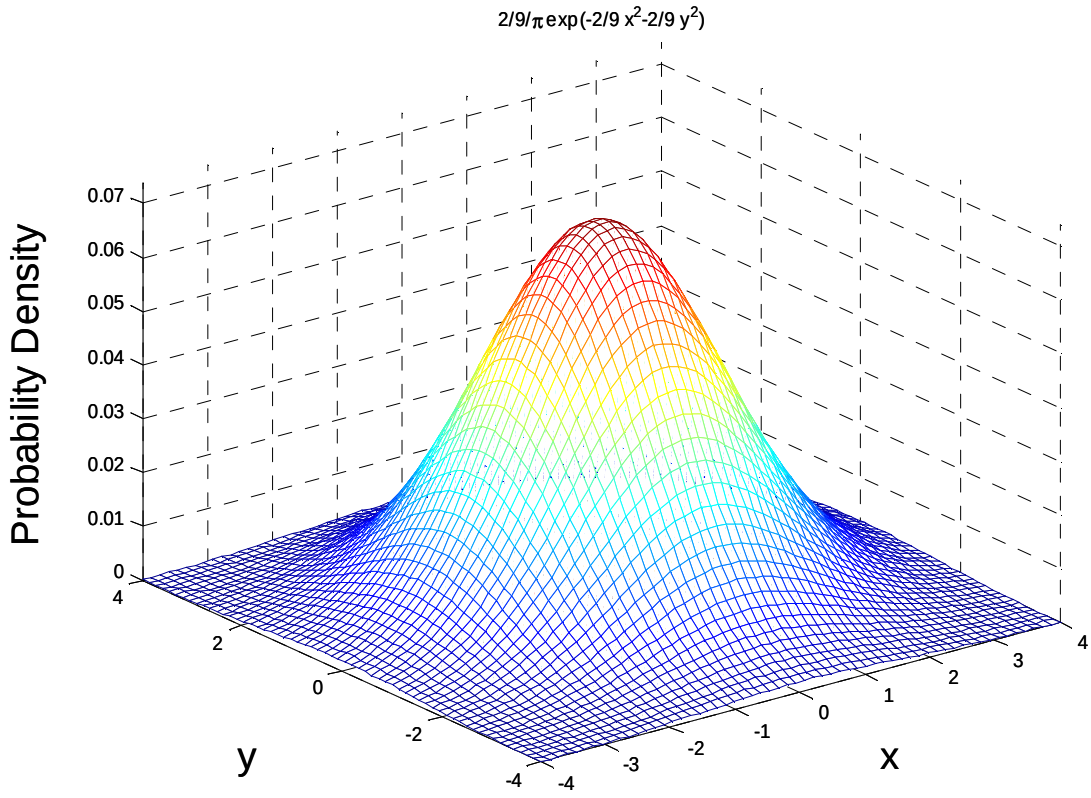


**Figure 7:** Joint PDF of Two Normal Random Variables

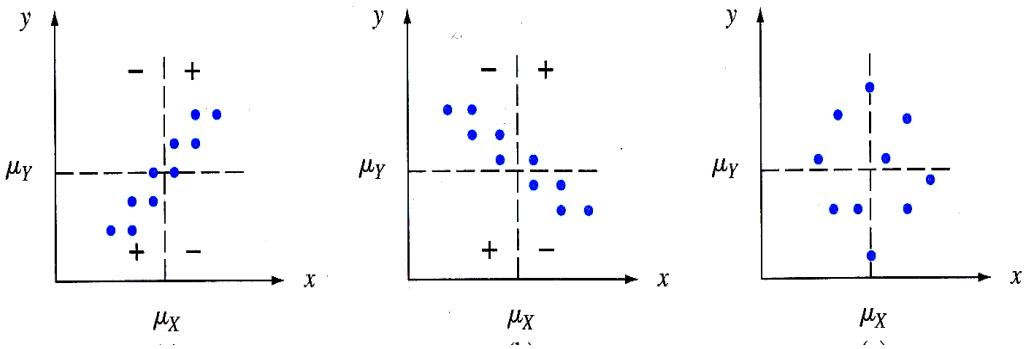


**Figure 8:** Illustration of Correlation Coefficient Measure [30].

covariance of two variables between $-1$ and 1 for easy comparison to other variable pair correlation. The correlation coefficient is calculated by dividing the covariance of two variables by the product of their standard deviations as follows

$$\rho_{x_i,x_j} = \frac{COV(X_i, X_j)}{\sigma_{x_i}\sigma_{x_j}} \tag{6}$$

This concept is illustrated in Figure 8 where the first graph on the left shows strong positive correlation near +1, the middle graph shows strong negative correlation near -1, and the last graph shows a data set where the correlation and hence the linear statistical dependence is near zero due to the scatter and obvious lack of a relationship between the variables.

The probabilistic terms already mentioned are for the case of a multivariate probability distribution, which may or may not be independent. An equally important problem is to calculate the probability of a function of random variables.

### 2.2.2 Functional Relationships

When an explicit relationship between a response and one or more random variables is required, the probabilistic analysis becomes even more complicated. For instance, the expected value of a general function of multiple random variables would then be calculated by substituting the random variables with a function of random variables.

The expected value or mean would then be calculated as

$$E[h(x_1, x_2, \ldots, x_n)] = \int_{-\infty}^{+\infty} \ldots \int_{-\infty}^{+\infty} h(x_1, x_2, \ldots, x_n) f(x_1, x_2, \ldots, x_n) dx_1 dx_2 \ldots dx_n \tag{7}$$

where $h(x_1, x_2, \ldots, x_n)$ is an arbitrary mathematical function of one or more variables and $f(x_1, x_2, \ldots, x_n)$ is the joint probability distribution function.

Not surprisingly, for most problems an explicit solution of the probability integral is very difficult, if not impossible, to obtain. In fact, there are only a handful of explicit solutions for the probability density function of known functions [45]. Some of the more popular solutions are for

- Sums and differences of Independent Normal Variables
- Products and Quotients of Independent Lognormal Variables
- Sums of Independent Poisson Random Variables

Exact probabilistic distributions of functions differing from these few cases are difficult to obtain. However, useful information about the properties of these distributions is realizable using reliability techniques discussed in the following section.

## *2.3 Structural Reliability*

Several approaches for performing probabilistic analyses have been developed over the years to determine the structural reliability of critical components involving a function of random variables. These methods are scattered across the literature over many applications. However, they can be categorized, for comparison, into two major classes: simulation and analytical.

### 2.3.1 Simulation Approaches

The simulation class of approaches involves generating statistical and probabilistic information through repeated evaluations of a computer model representing a real

process under random input. The most popular and most widely used simulation technique is Monte Carlo Simulation. Toucher [92] gave the first complete account of this method. It involves the following steps: assign a parametric probability distribution to each simulation input variable, generate random numbers from a uniform distribution ranging from zero to unity, convert the random numbers into actual variable values using the inverse cumulative distributions of the variables, compute corresponding response variable, and then repeating this process for a specified number of simulations. The calculation of the probability of a certain event occurring, such as component failure, is performed by applying a counting technique to the population of simulation results. This method is extremely appealing since it requires the least amount of statistical knowledge and asymptotically converges to an exact answer as the number of simulations approaches infinity. The number of simulations required to provide an accurate probability calculation is the primary limitation of simulation which is why low probability calculations require a large number of simulations.

A sub-class variant of Monte Carlo, called variance reduction techniques (VRT), was developed to overcome the sample size limitation of Monte Carlo. The primary approach of these techniques is to focus on a small region of the probability space. Monte Carlo simulation covers the entire probability space to generate the complete cumulative distribution function of the response of interest. The term variance reduction refers to the reduction of the dispersion of the estimated response without disturbing the estimated response mean.

VRTs can be grouped into three categories [16]: sampling methods, correlation methods, and other methods. The sampling methods control the focus of the simulation through constraints or distortions. Some of the sampling methods include systematic, stratified, importance, latin hypercube, and randomized. Sampling methods can drastically reduce the number of simulations required to find a single-point probability value. A thorough account of variance reduction techniques has been provided

by Mahadevan [62].

### 2.3.2 Analytical Approaches

The analytical class of probabilistic methods uses first and second-order approximations of the deterministic response to aid in evaluating the probability integral of equation (2). Approximations are determined using a power series expansion technique such as the following Taylor series expansion about the mean input vector, $\overline{X}$, given as

$$\begin{aligned} Z = g(X) &= g(x_1, x_2, \ldots, x_n) \\ &\approx g(\overline{X}) + \sum_{i=1}^{n} \frac{\partial g}{\partial X_i}(X - \overline{X})^T + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 g}{\partial X_i \partial X_j}(X - \overline{X})(X - \overline{X})^T + \ldots \end{aligned} \tag{8}$$

where $n$ is the number of input variables.

An early analytical method called First Order Reliability (FORM), which was initially formulated by Cornell [21], uses a first-order Taylor series expansion [7] approximation of the response function about the mean variable vector and makes use of second-moment statistics, namely mean and covariance. The resulting approximation of the response mean and variance[2] become

$$\mu_Z \approx g(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n) \tag{9}$$

and

$$\sigma_Z^2 \approx \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial g}{\partial X_i}\frac{\partial g}{\partial X_j} COV(X_i, X_j) \tag{10}$$

with a reliability safety index computed as

$$\beta = \frac{\mu_Z}{\sigma_Z} \tag{11}$$

[2]Refer to section A.1 for a derivation of the response variation

which, assuming that $Z$ is normally distributed, can be used to compute the probability of failure by simply using

$$p_f = \Phi(-\beta) \tag{12}$$

where $\overline{X}$ is the mean vector and the value of $\Phi(-\beta)$ can be found using any standard normal probability table. However, the reliability index given by equation (12) suffers from the fact that it is non-constant under different formulations of the same mechanically equivalent performance function. The Advanced First Order Second Moment method (AFOSM) proposed by Hasofer and Lind [46] uses a modified reliability index, $\beta_{HL}$, which is invariant with respect to different formulations of the response function.

Even with these improvements, first order methods can yield inaccurate reliability predictions should the input variables be non-normally distributed. Therefore, Rackwitz and Fiessler [74] developed an approach that transforms non-normal variables into equivalent normal random variables. The resulting Rackwitz-Fiessler transformation algorithm has seen much attention in the literature. Also, since the transformation approach assumes statistical independence between the input variables, techniques to uncorrelate these variables were developed such as the Rosenblatt transformation [77]. However, normalization or uncorrelation transformations can lead to a highly non-linear probability space which can be difficult to approximate with a linear approximation. An improved approach using Second Order Reliability Methods (SORM) was then taken which utilizes second order Taylor series expansion approximations of the response as a way to estimate the response mean and variance (Fiessler et al. [36], Breitung [19], and Der Kiureghian et al. [28]). Overall, higher order analytical methods can achieve greater accuracy yet they are difficult to implement due to their complexity and need for more partial derivatives.

Although the studies cited here are only a subset of an increasingly popular field, a sound statistical characterization of an $n$-dimensional local parameter space has yet

to be conducted. Without such a characterization, efficient and, more importantly, accurate probabilistic analyses of these components are not likely.

### 2.3.3 Dependency Considerations

Early probabilistic analyses addressing local parameter statistical dependency were conducted by Newell et al. [70]. These researchers recognized that reliability assessments of such a component would require an advanced integrated environment accounting for most if not all of the areas illustrated in Figure 3. They also recognized the complexity, not only of the multi-disciplinary deterministic analyses, but especially of the statistical behavior of the local parameter space. The approach taken to handle independent load parameters is that of the Gaussian Moment Method which assumes that each of the parameters is normally distributed. Wallace and Mavris [93] have created a relatively simple integrated gas turbine blade probabilistic analysis model that was recently used to demonstrate that several local parameters were in fact non-normal and therefore should be verified before making a normality assumption. Newell also found that some of the local parameters within their study were highly correlated. Due to the computational limitations of the late 80's when this work was conducted, the dependency of these parameters wasn't directly accounted for. Instead, they devised a method called the 'marginal distribution method' which used a simple gain coefficient matrix. The marginal conditional distribution for each dependent parameter is approximated using a linear function, the product of an $i^{th}$ independent parameter and a 'gain' coefficient.

Another equally important concept involved with component reliability is that of modeling multiple, concurrent mechanisms of failure. For components of complex systems such as hot gas path components of gas turbine engines, there is a multitude of failure phenomena that exist for a suspect component. Acknowledging that several failure mechanisms can contribute to the chance of failure of the part of interest,

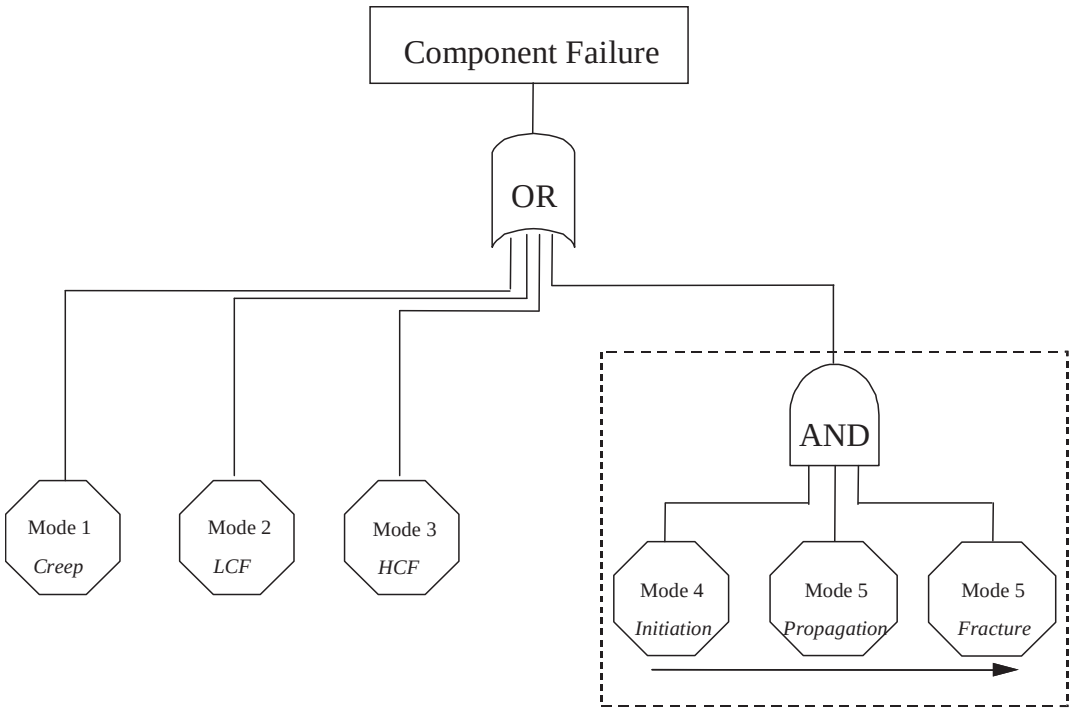


**Figure 9:** Component Fault Tree Analysis Accounting for Multiple Failure Modes.

how then does one account for this probabilistically? Many approaches have surfaced within the literature ranging from theoretical statistical approaches to those of a more applied nature which have seen a more widespread use.

Fault Tree Analysis (FTA) is a popular practical oriented approach used to model these competing failure mechanisms. Two or more failure events can be described with a Boolean structure comprised of series (OR) and parallel (AND) segments as shown in Figure 9. The process of calculating the probability of the top event, or failure probability, involves using set theory as shown by the equivalent graphical representations of a series system or its equivalent reliability block diagram in Figure 10. Here, the occurrence of any of $N$ events would cause the top event, failure, to occur. The event probability space equivalent of Figure 10 can be illustrated using a Venn event diagram shown, for events $A$ and $B$, by Figure 11. Each event is depicted by the areas conscribed by the ovals while the complete *set* of events is included within the rectangular area shown by the outer border of the Figure. The probability

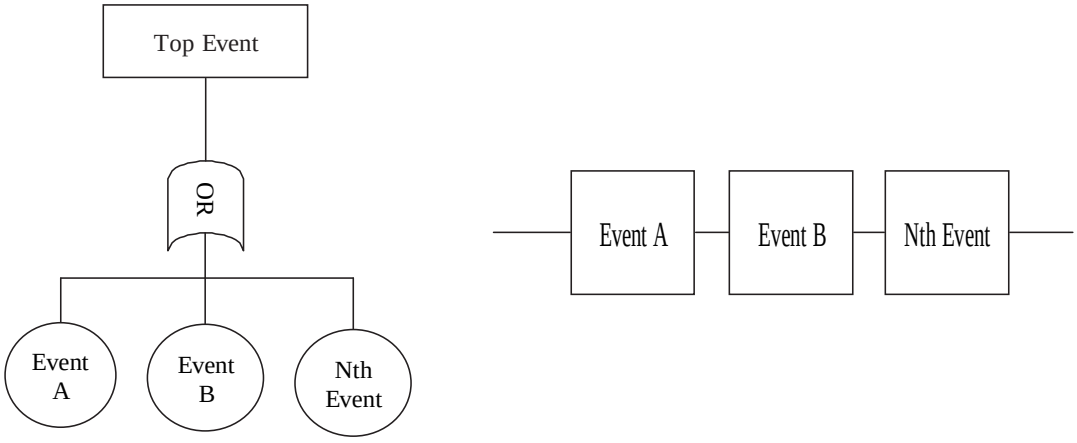


**Figure 10:** Fault Tree and Equivalent Reliability Block Diagram for a Series or 'Weakest Link' System.

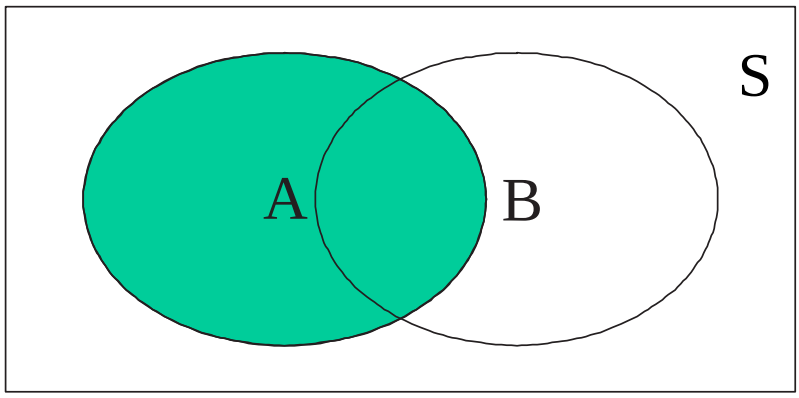


**Figure 11:** Venn Probability Event Diagram.

of the event occurring then would be proportional to its relative area shown on the Venn diagram. The overlapping area between the two events denotes the intersection or Boolean *AND* of the two events. If the two events were mutually exclusive, there would be no overlapping area, and thus no chance that both events could share the same event space. Mutual exclusiveness is not to be confused with independence of the events as independent implies a mutual, overlapping space.

The corresponding formulas used to calculate the event probability are simplified by assuming that the bottom events, or failure events, are independent. The result is a significant simplification to the system failure probability calculation. When the system failure condition requires any of several events to take place, a series system approach is used. If the system failure condition occurs when all of a set of events take place then a parallel system structure is present. The series system probability of failure can be calculated using a straightforward formula given as

$$P_f = P(E) = P(A \cup B \cup \ldots \cup N) = 1 - \prod_{i=1}^{n}(1 - P_i) \tag{13}$$

where $E$ is the failure condition event under consideration while $A$, $B$, through $N$ denote the individual events leading to the failure event, and $P_i$ is the i$^{th}$ event probability. For a parallel system probability of failure

$$P_f = P(E) = P(A \cap B \cap \ldots \cap N) = \prod_{i=1}^{n} P_i \tag{14}$$

both of which are generalized for $N$ events or in the case of component reliability, failure modes, and their corresponding failure probabilities. Parallel failure conditions are the most ideal as the failure probability of a parallel configuration is always less then that of a series system for the same event probability values.

FTA was initially applied to structural reliability by Madeson [61], and has since seen much use in modeling multiple failure modes of single components of complex systems. However, this method has some strong limitations in that the underlying statement for calculating the probability of the top event relies on a rather restrictive

assumption of independence between each of the failure mode, or bottom, events. An indirect technique to address this issue uses bounds to compute an interval within which the actual probability of failure will lie. Dependent bottom events can be accounted for using first or second order bounds on the actual 'joint' probability of failure calculation. First order bounds (Cornell [21]) are simply

$$max[P(E_i)] \leqslant p_{f_s} \leqslant min\left[\sum_{i=1}^{n} P(E_i), 1\right] \tag{15}$$

where $P(E_i)$ is the probability of failure of the i$^{th}$ failure mode and $n$ is the number of failure modes. The first order bounds can be quite large so second order bounds are more widely used. Second order bounds (Ditlevson [31]) can be calculated using

$$\begin{aligned} &P(E_i) + \sum_{i=2}^{n} max\left\{\left[P(E_i) - \sum_{j=1}^{i-1} P(E_iE_j)\right], 0\right\} \\ &\leqslant p_{f_s} \leqslant min\left\{\left[\sum_{i=1}^{n} P(E_i) - \sum_{i=2,j<i}^{n} max(P(E_iE_j))\right], 1\right\} \end{aligned} \tag{16}$$

where $P(E_iE_j)$ is the joint probability of the i$^{th}$ and j$^{th}$ events. The joint probability of two events is difficult to find exactly, especially for nonlinear limit-states. There are several methods in practice that can be used to approximate joint probability of two events such as that proposed by Ditlevson [31] for linear, limit-state functions and the approach of Rackwitz and Fiessler [74] which is ideal for nonlinear event functions. Finally, there are simulation based techniques such as those recommended by Bandte et al. [11] and Nataf [69] that can be applied to remedy this situation and maintain the accuracy of the analysis. These are discussed in Chapter 3.

The results obtained using the FTA method are single probability values which are in fact a point on the cumulative distribution of a single composite failure probability curve. A more general approach to FTA would be to model the various failure modes with a failure distribution and solve for the top-level event failure distribution as opposed to an individual probability value. Such is the approach taken using

the competing risks theory (David and Moeshberger [25], Birnbaum [14], and Lawless [52]). This approach utilizes the marginal distributions of each failure mode to construct a composite failure distribution, also called the net life, of the item under the assumption that the failure modes are independent. The joint failure probability density function for independent, series-event failure modes is defined as

$$f_{s_j} = \frac{\pi_j f_{Y_j}(x)}{\sum_{i=1}^{n} \pi_i \left(1 - \int_{-\infty}^{x} f_{Y_i}(x)\right)} exp\left(-\int_0^x \frac{\pi_j f_{Y_j}(x)}{\sum_{i=1}^{n} \pi_i \left(1 - \int_{-\infty}^{x} f_{Y_i}(x)\right)}\right) \tag{17}$$

where $\pi_j = \int_0^\infty -\frac{\partial}{\partial Y_j}\left[1 - \int_{Yn}^{\infty} \ldots \int_{Y_1}^{\infty} \prod_{i=1}^{n} f_{Y_i}(x) dY_1 \ldots dY_n\right]_{Y_1=\ldots=Y_n=Y}$ and is the probability of failure from mode $j$. Interestingly, this formula combines any required combination of failure distributions, whether each are normal or not, and thus is quite flexible. However, this formula is a density function and therefore must be integrated to find the cumulative density function for probability calculations. Also, it is limited to independent, series event failure modes. Unfortunately, failure modes and input variables can be statistically dependent. Modelling this dependence is extremely difficult, yet important, and requires the use of joint probability theory which is discussed in Chapter 3.

## *2.4 Statistical Reliability*

Structural reliability, discussed earlier, is an area of probabilistic approaches is focused on developing the mathematics around the actual physical analysis and is therefore more of an early reliability prediction class of methods before experimental data is available. A different approach was in use long before structural reliability methods were developed. This approach is from the statisticians point of view, as it neglects the physical reasoning of the problem at hand and focuses on letting the data, failure data in this application, lead the analysis. The data, historically has been generated through experimental or actual operational experience. Now that engineering analysis through computer simulation has become more widespread, statistical techniques

using simulation based data can potentially improve the early prediction of component reliability. Within the statistical reliability field, there are several methods that are important to the objectives of this thesis study. They are methods for identifying a univariate distribution, creating multivariate distributions, and combining distributions using a mixture model.

### 2.4.1 Univariate Distribution Identification

As described earlier, the direct solution for the probability density function of response which is an arbitrary function of several random variables is difficult to produce. A few exceptions include those specific cases that are listed earlier as well as known distributions of a multivariate function through apriori statistical evidence. For example, previous statistical evidence has shown generally that the probability density function of the creep failure of a metallic alloy at high temperature is log-normal. Many distributions of failure functions related to crystalline materials can be also modelled with another two-parameter distribution known as the Weibull distribution (Weibull [97]). Fox [37] has compiled a list of regularly used distributions used in physics-based reliability assessments including typical response (life and failure) and input (geometric, tolerances, material defects, and modal uncertainty) univariate distributions. But which distribution should be used for each failure mode? And, what values should be specified for the distribution parameters?

The determination of the parameters of the distribution are accomplished using a parameter estimate technique such as maximum likelihood theory, method of moments, or graphical procedures (Fox [37]). Identifying a distribution is more involved. First, a set of candidate distributions must be selected and their respective parameter values determined using a parameter estimation technique. Then, a test statistic is

computed using one of several available methods such as the popular Kolmogorov-Smirnov test (Lawless [52]) to test for the null-hypothesis that the proposed distribution does not fit the empirical data using a specified significance level. Assuming that more than one distribution results in accepting the distribution, a relative ranking statistic such as Anderson-Darling can be used to determine which of the potential distributions predicts the data more accurately (Anderson and Darling [6]). This statistic is particularly useful for comparing distributions with unique tails such as the normal, log-normal, and Weibull distributions since it weights data points within the tail regions of the distribution heavier. The distribution as well as the estimates of the distribution parameters is now available.

Two methods that have been used for identifying random variable distributions are the Kolmogorov-Smirnov and Anderson-Darling EDF[3] based methods. For each method, a set of data, such as component life times, is first compiled or generated. The Kolmogorov-Smirnov (K-S) goodness-of-fit test is useful in determining whether or not a given data set follows a hypothesized, continuous distribution. The K-S test is quite easy to implement yet completely general for any suspect parametric distribution given that the empirical distribution[4] is available. The K-S test statistic is defined as the maximum difference between the empirical cumulative distribution function $\hat{F}(x)$ and the hypothesized fitted cumulative distribution function $F_o(x)$.

The null and alternative hypotheses for this test are:

$$H_o : F(x) = F_o(x)$$
$$H_a : F(x) \neq F_o(x)$$

The test statistic for a complete data set is

$$D_n = sup_x|\hat{F}(x) - F_o(x)|$$

[3]EDF refers to Empirical Distribution Function which is a non-parametric determination of the cumulative failure (or survival) function which is the basis for many distribution test methods.

[4]The empirical distribution is created using the given data set

where sup is an abbreviation for supremum. Thus, the test statistic is computed by looping over all of the data. Larger values of $D_n$ indicate a poorer fit of the hypothesized distribution. The test statistic formula can be given in a slightly different form for computational efficiency as

$$D_n = max\left\{max_{i=1,2,\ldots,n}\left(\frac{i}{n} - F_o(x_i)\right), max_{i=1,2,\ldots,n}\left(F_o(x_i) - \frac{i-1}{n}\right)\right\}$$

where $i$ is the $i^{th}$ failure, $x_i$ is the $i^{th}$ sequential lifetime or response value and $n$ is the total number of realizations (samples) of the random variable. Algorithmically, one would fail to reject the null hypothesis and accept the hypothesized distribution with significant evidence if the computed K-S test statistic is less than the critical value of the statistics. The critical value is independent of the hypothesized distribution which is an appealing characteristic of this method. However, the K-S goodness of fit test is only valid for fully specified distributions. Therefore, if the same data for the goodness-of-fit test was used to estimate the distribution parameters then the K-S statistic method cannot be used. Due primarily to this limitation, the Anderson-Darling test statistic is recommended.

The Anderson-Darling (A-D) test is a modification of the K-S test that weights the tails of the distribution and utilizes the hypothesized distribution resulting in a more sensitive test for goodness of fit. Contrary to the K-S statistic, the A-D test uses the hypothesized distribution to calculate the critical value of the test statistic. Although superior to the K-S statistic, the A-D method requires critical values to be computed unique to each distribution. Further, tables of critical values found in the literature have only been created for a few distributions such as the normal, lognormal, exponential, Weibull, extreme value type I, and logistic distributions. Fortunately, these distributions are some of the most widely used in statistics, especially in reliability modeling.

The A-D test statistic is defined as

$$A^2 = -n - S$$

where

$$S = \sum_{i=1}^{n} \frac{(2i-1)}{n} \left[ ln\left(F_o\left(x_{(i)}\right)\right) + ln\left(1 - F_o\left(x_{(n+1-i)}\right)\right)\right]$$

In order to compare the A-D test statistic to its critical value, it must be modified by a factor which is dependent on the number of samples as well as the specific distribution proposed. For example, assume the case of testing the goodness-of-fit of a normal distribution for a given data set. If the location and scale (mean and variance) are to be estimated from the same data as that used for the test, then the appropriate modified A-D test statistic will be

$$A^2 = \left(1 + \frac{4}{n} - \frac{25}{n^2}\right)$$

The A-D method without using the critical value estimate has in the literature been applied to several distributions simultaneously. In this application, the distribution with the smallest A-D test statistic value would be the most appropriate distribution of the set to model the data. Once the most ideal distribution is selected, the critical value could then be calculated to statistically test for a given significance level whether or not the distribution is truly representative of the data set. See the work by Stephens for a list of modifying factors and critical test statistic values as well as a thorough account of goodness-of-fit test techniques (Stephens [82][83][84][85][86]).

### 2.4.2 Multivariate Distribution Modeling

The cumulative probability density or equivalent survivability (reliability) functions that have been discussed thus far assume that the given operating conditions are statistically invariant. However, it can be argued that the system statistical characteristics can vary during the design process and even after the system has entered

service as its usage is changed. Differing statistical inputs result in altering the failure probability function of the component and therefore can lead to vastly different reliability values. There are two approaches for capturing this behavior that can be found in the literature. First, the individual point probability value for a given response level, or conversely the individual response level for a given probability value can be considered a response and some sort of mathematical model can be created to account for the effect of the input characteristics [27]. Second, the underlying failure distribution can be extended from a univariate distribution to one involving multiple variables through the use of a covariate model.

Covariate theory models are used to model the affect of suspect treatments in a lifetime or reliability model. A covariate is defined as a treatment or explanatory variable that influences the failure time of the component or item. A vector of covariates, $z$, is chosen with each entry of the vector representing a unique explanatory variable. Typical covariates include those that represent mechanical forces, material properties, and environmental variables. Two rather popular approaches for linking these covariates to the probability function are based on either modifying the survivor function[5], also known as the accelerated life model, or by modifying the hazard rate function, known as a proportional hazards model. The covariate model is created by modifying the failure distribution function with a 'link' function, $\psi(z)$, which is a function of the covariates. The function commonly chosen is the log-linear link function given as

$$\psi(Z) = e^{\beta' z} \tag{18}$$

where $\beta$ is an $n$ by 1 vector of regression coefficients for each of the $n$ covariates in the vector, $z$. The appealing characteristic of this link function compared to others that have been proposed is that the exponential form is highly compatible with most

[5]The survivor function is the complement of cumulative failure probability function given by $S(t) = 1 - F(t)$ where $F(t)$ is the cumulative failure probability function

traditional parametric distributions. Also, the log-linear function is stable across large ranges of the regression coefficients especially asymptotically.

The resulting survivor function in the accelerated life testing (ALT) approach becomes

$$S(t) = S_o(t\psi(z)) \tag{19}$$

where $S_o$ is the baseline survivor function determined at the nominal values of each of the potential covariates (i.e. $z_i = 0$). This formulation imposes certain requirements on the link function. Namely, the function must be equal to unity at the nominal covariate setting, $z = 0$, and must be positive for any and all values of $z$. In essence, the ALT approach uses the link function to modify the value of the random variable. For a life distribution this amounts to a scalar modifier of the time axis. Finally, the form of the log-linear link function used in both of the two approaches allows for the use of the more familiar regression model theory to determine the respective covariate regression coefficient coefficients, $\beta_i$.

Several papers have been authored over the past few decades which provide the detailed theory behind this approach including techniques used to simultaneously estimate the baseline parameters and the covariate regression coefficients using survival data points [53]. Further, Singpurwalla has applied covariate theory to proposed probabilistic models of the reliability of components operating in dynamic environments (Singpurwalla [80]). In this approach he proposes treating the covariates as stochastic processes and has found a few rather interesting albeit limiting probabilistic failure models of an explicit nature for certain specified conditions. However, the mathematical complexity of the resulting covariate-process models for more general cases hampers their usefulness.

### 2.4.3 Mixture Distributions

In practical reliability problems, the lifetime data of the sample to be analyzed might be of different sources. Should the distributions of the individual subsets of lifetime data differ, it may be difficult to find or fit an existing parametric distribution. A remedy to this problem is to use a mixture distribution.

A mixture distribution is a distribution that is composed of multiple distributions each representative of unique lifetime populations and is modeled as

$$f(y) = \int_{all\theta} f(y|\theta)p(\theta)d\theta \tag{20}$$

where $\theta$ is called the mix parameter and $p(\theta)$ is its distribution. A discrete distribution, which is usually the case for heterogeneous populations, would require $p(\theta)$ to be a probability mass function. Therefore a finite mixture distribution would take the form of

$$f(y) = \sum_{i=1}^{m} p_i f_i(y|\boldsymbol{\theta}_i) \tag{21}$$

where $f_i(y|\boldsymbol{\theta}_i)$ is the density function of the $i^{th}$ population, $\boldsymbol{\theta}_i$ is the vector of distribution parameters for $i^{th}$ population density function, and the $p_i$ are, of course, the mixture parameters whose sum over all the populations must be equal to unity. These mixture parameters must be non-negative as required for any probability density function. This particular model has been identified within this research as potentially a very useful tool to address the mismatch in the source of uncertainty between the various random variables that would be considered in a component reliability analysis.

## *2.5 Applied Statistical Models*

A key area that will facilitate the development of the proposed thesis work is that of applied statistical models. These models are envisioned as complementary techniques

used to create simpler disciplinary analysis models that can permit the implementation of the proposed probabilistic methods. In short, approaches within this field have been created that merge traditional regression techniques with statistical theory to enable the construction of superior models that are sound approximations to real world phenomena, or in this case complex modeling and simulation modules. A particularly useful area within this field is that of Response Surface Methods (RSM). The RSM method determines an appropriate fit of an assumed function about samples of a response to create an explicit functional representation of a more complex physical model. The Design-Of-Experiments method (DOE) is employed to select an appropriate combination of variable settings to efficiently sample the actual response space. Statistical measures are then taken to gain a more quantitative understanding of the actual response space and created potential predictive models of that space. These methods were first introduced by Box and Wilson [18] and were then developed to a more useable form by Box and Hunter [17]. The DOE typically used is a three level central composite design, shown in Figure 12, which permits the modelling of interactions between several of the main factors as well as quadratic main effects. A representative quadratic, polynomial metamodel[6] commonly used is given as

$$R = b_o + \sum_{i=1}^{k} b_i x_i + \sum_{i=1}^{k} b_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=1+1}^{k} b_{ij} x_i x_j + \epsilon \tag{22}$$

where $b_i$ are regression coefficients for the first order terms, $b_{ii}$ are the coefficients for the pure quadratic terms, $b_{ij}$ are the coefficients for the first order cross-product terms, $x_i$, $x_j$ are the design variables, and $\epsilon$ is the error term vector of which its components are assumed to be independent and normally distributed with constant variance.

This particular function belongs to the class of response surface equations (RSE)

[6] A metamodel is a more tractable model created from other usually more complicated models

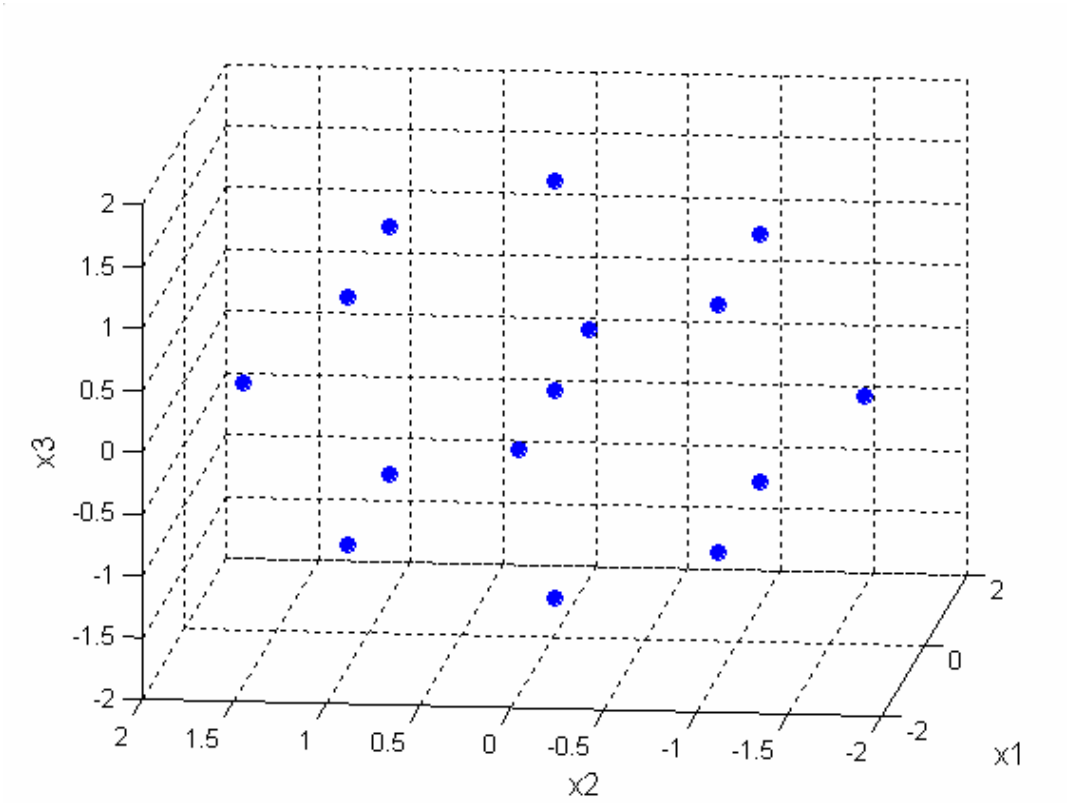


**Figure 12:** Central Composite Design Sample Point Scheme.

which are multivariate closed form expressions and of which its coefficients are determined through regression using a representative response sampling. The sampling can be accomplished using a balanced DOE which permits numerous statistical analyses to be performed to check the accuracy, validity, and usefulness of the RSE for both understanding the sampled response and especially for predictive purposes. The predictive capability of RSEs is paramount to the proposed research since complex, time-consuming analyses can be replaced by such an explicit function. This is extremely beneficial when probabilistic analyses are conducted since numerous evaluations can be performed over a short amount of time. However, RSEs are applicable to problems with only a few variables. Screening approaches based on DOE or sensitivity analyses can be used to reduce the set of variables to those that are the strongest drivers of the response of interest. In addition, RSEs have been applied by Koch et al. [51] in a hierarchical fashion to create a metamodel of an entire commercial turbofan engine model requiring the use of several variables.

## *2.6 Summary: Probability and Engineering*

Several corporate initiatives within many companies have attempted to incorporate statistical and probabilistic techniques within their design and development processes for assessing the reliability of components such as gas turbine blades. However, these techniques appear to be more of a management science approach and may fail to directly address unique statistical phenomena associated with the analysis of components of complex systems. O'Connor [71] addresses these concerns as well as many others such as the increased uncertainty in predicting the tail ends of failure distributions as well as the crutch of working with reliability as just a haphazardly generated number that is amenable to misuse and misunderstanding. O'Connor summarizes the current widespread use of reliability, in general, as misleading and worthy of considerable revamping. Several other experts including Bier [13] and Golomski [43] agree that

reliability can and has historically been misused. In accordance with O'Connor's recommendations, conducting probabilistic analyses of components of complex systems requires a more sound and structured statistical posture than that which has been developed and executed thus far. The background theory discussed in this chapter has been identified as a proper starting point for this research. Improving the process of conducting reliability assessments of components of complex systems via computer simulation is explored within this study. Barriers to such a posture for reliability assessment from simulation will be identified and remedial measures proposed and developed.

# CHAPTER III

# JOINT PROBABILITY MODELS

There are many situations in practical probabilistic analyses where more than one random variable is of interest. This condition is especially true for component reliability approaches where there are several potential random variables across several categories namely geometric, material, and loading. Simplifying yet potentially inappropriate statistical assumptions are often made for this multiple parameter space. Given the focus of the research to improve the statistical input necessary for component reliability assessments, it is imperative that several methods of modelling multivariate probability, i.e. joint randomness, be considered and evaluated. Before such methods are discussed, a bar reliability example using the principles described in section 2.3 is given to demonstrate the need for modeling joint randomness.

## *3.1 Bar Reliability Example*

Subsequent sections are to offer several approaches that can model joint randomness of the space of interest. But first, a bar reliability problem is given to illustrate the importance of modeling joint randomness during reliability assessments. First, independence of the bar stress and strength responses is assumed and the probability of failure determined. Then, the actual problem solution is found by modeling the physics-driven statistical dependency.

### 3.1.1 Problem Statement

The problem statement is to compute the probability of failure of a fixed-fixed bar, shown in Figure 13, made of Inconel-100 undergoing thermal contraction. The reliability solution is to be found using the stress versus strength limit-state approach

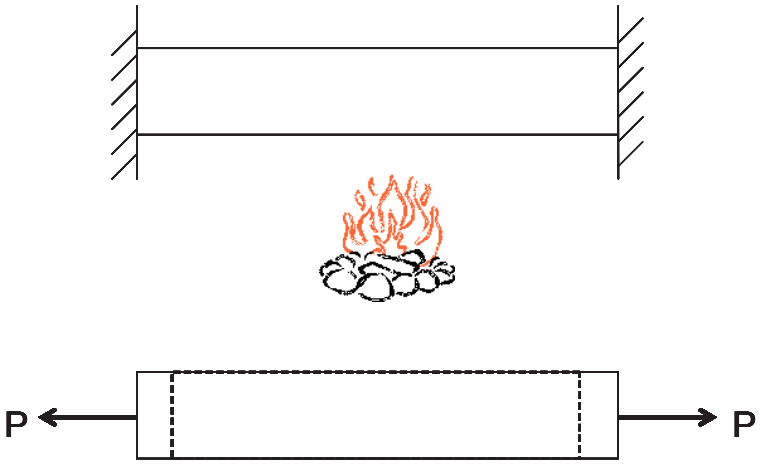


**Figure 13:** Fixed-Fixed Bar Under Thermal Contraction.

described in section 2.3. If the bar is clamped or fixed after it has been heated to some reference temperature and the temperature is reduced from this reference temperature, a negative thermal strain (i.e. displacement) is produced. Since the bar is fixed on either side, static equilibrium and Hooke's law requires that a tensile reaction loading be present during cooling resulting in a tensile stress imposed on the bar. Failure occurs when the imposed tensile stress exceeds the rupture strength of the bar. Finally, the bar temperature and bar reference temperature at clamping are assumed to be independent normal random variables. Since these variables are random, the strength and stress responses will also be random variables.

### 3.1.2 Physical Model

The physics of this problem can be represented as follows. Assuming that the change in temperature is small enough, the thermal strain can be expressed as

$$\epsilon_{th} = \alpha \Delta T = \alpha \cdot (T - T_{ref})$$

where $\alpha$ is the thermal expansion coefficient, $T$ is the bar temperature, and $T_{ref}$ is the reference temperature at which the thermal strain is zero. The thermal stress, $S_{th}$, can then be solved using static equilibrium and is given as

$$S_{th} = -E\alpha \cdot (T - T_{ref}) \tag{23}$$

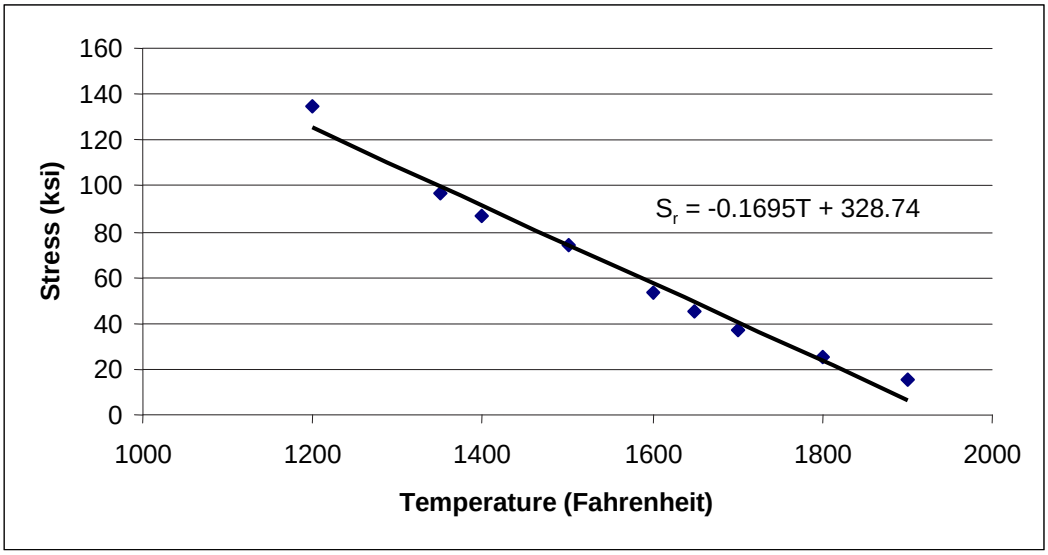


**Figure 14:** In-100 Temperature Dependent Rupture Strength Data [66].

where $E$ is the elastic modulus. When the actual temperature, $T$, is less then the reference temperature, $T_{ref}$, the change in temperature is negative resulting in a positive or tensile thermal stress. Tensile thermal stress is the demand placed on the bar. The resistance of the bar to this demand is its stress rupture strength, which also is a function of temperature. Many models can be used to model the change in stress rupture strength with temperature; however, a linear model is assumed here for simplicity. The rupture strength versus temperature for a high temperature nickel alloy, Inconel 100, is shown in Figure 14. The relationship between rupture strength and temperature can be modelled using a linear function

$$S_r = a \cdot T + b \tag{24}$$

where $T$ is the material temperature, and $a$ and $b$ are the slope and intercept terms, respectively. The values for these constants are given in Figure 14. Values for all of the bar reliability properties are given in Table 1.

**Table 1:** Bar Reliability Input Properties.

| Property | Description | Units | Mean | Standard Deviation |
|---|---|---|---|---|
| $T$ | Beam Temperature | $^\circ F$ | 1700 | 10 |
| $T_{ref}$ | Reference Temperature | $^\circ F$ | 1800 | 10 |
| $\alpha$ | Expansion Coefficient | $\frac{1}{^\circ F}$ | 8.6e-6 | – |
| $E$ | Elastic Modulus | ksi | 31000 | – |
| $a$ | Coefficient | $\frac{ksi}{^\circ F}$ | -0.1695 | – |
| $b$ | Intercept | ksi | 329 | – |

### 3.1.3 Independence Solution

The common approach to component reliability is to assume independence of the stress and strength parameters. Therefore, this assumption is applied here to find the closed-form analytical solution to the failure probability of the bar. Starting with the limit state definition, the failure limit state is given as

$$g = R - D = S_r - S_{th} \tag{25}$$

where $g$ is the limit state or g-function defined as the resistance, $R$, minus the demand, $D$, which in this problem is the rupture strength, $S_r$, minus the thermal stress, $S_{th}$, given by equations (24) and (23), respectively.

The solution requires that the distribution of the g-function be found, which in turn requires that the distribution of both the demand and resistance also be found. By the theory of a linear sum of normal random variables, both the bar demand, $S_{th}$, and bar resistance, $S_r$, parameters are normally distributed random variables as their inputs are a linear function of normal random variables. Then by this same principle, the g-function is also a normal random variable. To fully describe these distributions requires that only the mean and variance be computed. The mean of any of these parameters can be found by simply substituting the mean of the input random variables into the functions. Variance, which is the square of the standard deviation, of the parameters can be computed using the analytical formula given by

equation (10). Using this equation, the demand variance[1] is given as

$$\sigma_D^2 = \sigma_{S_{th}}^2 = \left(\frac{\partial S_{th}}{\partial T}\right)^2 \sigma_T^2 + \left(\frac{\partial S_{th}}{\partial T_r}\right)^2 \sigma_{T_r}^2 \tag{26}$$

where $\sigma_T$ is the standard deviation of the bar metal temperature and $\sigma_{T_{ref}}$ is the standard deviation of the bar reference temperature. By evaluating the partial derivatives, equation (26) reduces to

$$\sigma_{S_{th}}^2 = (E\alpha)^2 \left(\sigma_T^2 + \sigma_{T_{ref}}^2\right) \tag{27}$$

which through substitution $\sigma_{S_{th}} = 3.77$ ksi. Applying equation (10) in a similar fashion, the resistance variance can be shown to be

$$\sigma_{S_r}^2 = a^2 \sigma_T^2 \tag{28}$$

which through substitution $\sigma_{S_r} = 1.70$ ksi. The mean and variance of the g-function can now be found where the g-function mean is given as

$$\mu_g = g(\mu_D, \mu_R) = \mu_R - \mu_D = \mu_{S_r} - \mu_{S_{th}} \tag{29}$$

and, by again applying equation (10), the g-function variance is given as

$$\sigma_g^2 = \left(\frac{\partial g}{\partial D}\right)^2 \sigma_D^2 + \left(\frac{\partial g}{\partial R}\right)^2 \sigma_R^2 \tag{30}$$

which reduces to

$$\sigma_g^2 = \sigma_{S_{th}}^2 + \sigma_{S_r}^2 \tag{31}$$

The mean, $\mu_g$, and standard deviation, $\sigma_g$, of the g-function can now be found through substitution as 14.19 ksi and 4.13 ksi, respectively. Finally, this result can easily be used with equation (12) to find the probability of failure[2] given as

$$P_f = P[g < 0] = P[S_{th} > S_r] = \Phi\left(-\frac{\mu_g}{\sigma_g}\right) \tag{32}$$

[1]The third term in equation (26) isn't included as it is a function of the covariance of $T$ and $T_{ref}$ which is zero due to statistical independence.

[2]Since the g-function is a normal random variable, the probability of it being zero can be found by simply finding the standard normal variable, $z = \frac{0-\mu_g}{\sigma_g} = \frac{-\mu_g}{\sigma_g}$, and then finding the respective cumulative probability value from a standard normal table.

which, using the independence assumption, is found to be 2.99e-4. However, this result is far from the actual solution when one considers the statistical dependence between the beam demand and resistance.

### 3.1.4 Actual Solution

The actual reliability solution of this bar problem can be solved by recognizing and computing the linear statistical dependence between the beam demand and resistance parameters; then, recomputing the bar failure probability. By inspection, functional dependence is expected as the bar stress and resistance, or strength, functions both are temperature dependent. Since the bar temperature is a random variable, statistical dependence will also be present and by the principle of a linear sum of normal random variables the statistical dependence is linear.

Statistical linear dependence can be quantified using the covariance measure introduced in section 2.2.1. Equation (105) derived in section A.2 can be used to express the covariance of the demand and resistance responses as

$$COV(R,D) = \frac{\partial S_r}{\partial T}\frac{\partial S_{th}}{\partial T}\sigma_T^2 + \frac{\partial S_r}{\partial T_{ref}}\frac{\partial S_{th}}{\partial T_{ref}}\sigma_{T_{ref}}^2 + \frac{\partial S_r}{\partial T}\frac{\partial S_{th}}{\partial T_{ref}}COV[T,T_{ref}] \qquad (33)$$

where $COV[T,T_{ref}]$ is the covariance between $T$ and $T_{ref}$. Since the material resistance, $S_r$, is not a function of the reference temperature, $T_{ref}$, and $COV[T,T_{ref}] = 0$ due to independence between $T$ and $T_{ref}$, the last two terms of equation (33) are equal to zero. Then by evaluating the remaining partial derivatives, equation (33) reduces to

$$COV(R,D) = -aE\alpha\sigma_T^2 \qquad (34)$$

which is computed to be $COV[R,D] = 4.52$ ksi$^2$. The correlation coefficient, a normalized measure of linear statistical dependence, can be computed by substituting the result of equations (34), (27), and (28) into equation (6) and is found to be $\rho_{R,D} = 0.71$. Thus, the bar strength and stress actually exhibit strong positive

correlation which should be accounted for in the probabilistic analysis of the limit-state g-function.

Since linear dependence is present between these two parameters, the g-function variance formula using equation (10) is actually a three term function given as

$$\sigma_g^2 = \left(\frac{\partial g}{\partial D}\right)^2 \sigma_D^2 + \left(\frac{\partial g}{\partial R}\right)^2 \sigma_R^2 + \underbrace{\left(\frac{\partial g}{\partial R}\right)\left(\frac{\partial g}{\partial D}\right) COV[R,D]}_{\text{Statistical Dependency}} \tag{35}$$

which reduces to

$$\sigma_g^2 = \sigma_{S_{th}}^2 + \sigma_{S_r}^2 - 2COV[R,D] \tag{36}$$

Then by substituting, the actual g-function standard deviation is found to be $\sigma_g =$ 2.84 ksi which is around two thirds of the variance predicted using the independence assumption. Finally, using equation (32) the actual failure probability is computed as $P_{f,actual}$ =2.85e-7 which shows that the solution obtained using the independence assumption, although conservative, is differs from the actual solution by orders of magnitude.

The effect of assuming independence of the g-function parameters is illustrated in Figure 15. The standard deviation, or dispersion, of the g-function distribution is greatly over-predicted when one uses the independence assumption. This result can be further explained, analytically, by comparing the actual g-function variance solution given by equation (36) with the solution found using the independence assumption given by equation (26). Because the third term in the actual bar reliability solution is neglected, although significant, the calculated variance is much larger using the independence assumption. Generally speaking, this behavior is of great importance to component reliability. By virtue of differing functional relationships of other problems, the dependence terms such as those of equation (36) could just as easily be positive *and* significant, and thus lead to an overly *unconservative* prediction using the independence assumption.

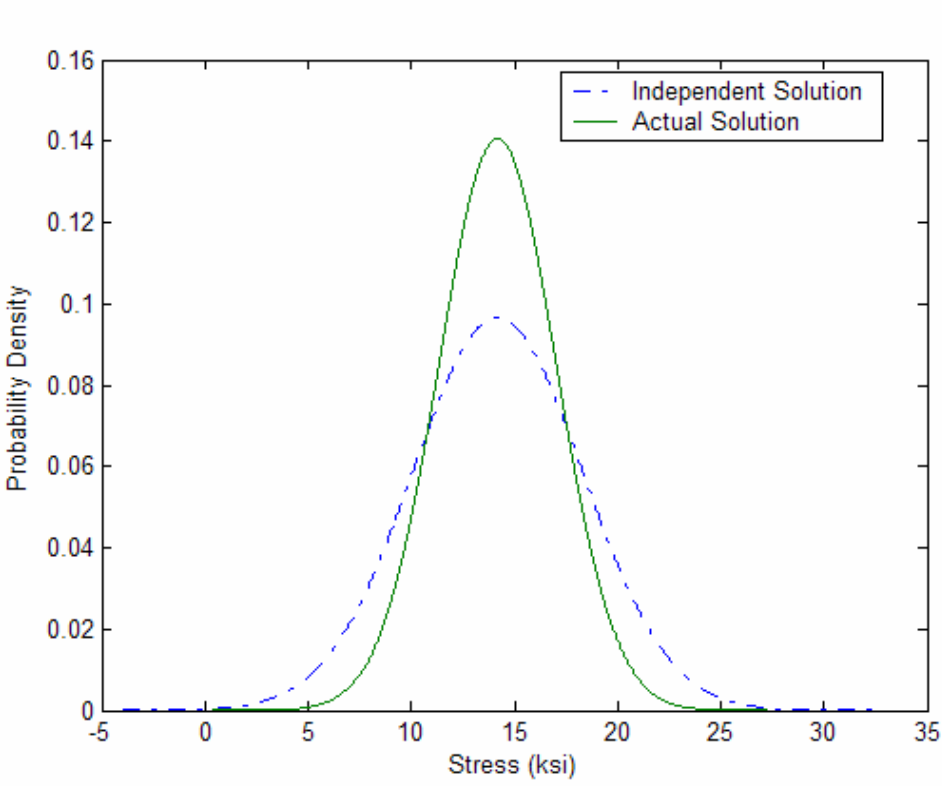


**Figure 15:** Effect of Independence on Bar g-function Distribution.

The results of this simple study suggest that considerable inaccuracy can be introduced during component reliability studies when neglecting input joint randomness. As shown later during the implementation phase of this study (Chapter 5), considerable joint randomness was present for components operating in complex systems. Therefore, the need for modelling joint randomness is apparent. The fundamentals of joint probability theory for two or more variables are now given.

## *3.2 Mathematical Foundation of Joint Probability*

The previous section demonstrated how the joint randomness of two normally distributed random variables can be modelled. This can be generalized to the case of multiple, arbitrarily distributed random variables. The general expression for the joint probability of several continuous multiple random variables is given by equation (2) and is repeated below.

$$P[(Y_1, Y_2, \ldots, Y_m) \in A] = \int_A \ldots \int_A f(y_1, y_2, \ldots, y_m) dy_1 dy_2 \ldots dy_m$$

where $f(y_1, y_2, \ldots, y_m)$ is the multivariate joint density function. This function has the following conditions:

- Positive definite: $0 \leqslant f(y_1, y_2, \ldots, y_m)$
- Integral property: $\int_\Omega \ldots \int f(y_1, y_2, \ldots, y_m) dy_1 dy_2 \ldots dy_m = 1$

Given a joint density function of multiple random variables, the univariate distribution function, called the marginal distribution, can be determined by integrating with respect to the other variables over the entire probability space. As an example, for a bivariate joint distribution function between random variables $Y_1$ and $Y_2$ the marginal distribution for $Y_1$ can be solved as follows

$$f_{X_1}(y_1) = \int_{-\infty}^{+\infty} f(y_1, y_2) dy_2 \text{ for } -\infty < y_2 < +\infty \tag{37}$$

The marginal distribution is of practical importance since most existing statistical characterizations involve that of the marginal distributions.

If the variables are statistically independent, then this greatly simplifies the formula for joint probability. The random variables $Y_1, Y_2, \ldots, Y_m$ are said to be independent if for *every* i$^{th}$ subset $Y_{i_1}, Y_{i_2}, \ldots, Y_{i_m}$ of the variables, and thus the joint probability function is equal to the product of the marginal distribution functions given as

$$f(y_1, y_2, \ldots, y_m) = \prod_{i=1}^{m} f_{Y_i}(y_i). \tag{38}$$

where $f_{Y_i}(y_i)$ is the marginal distribution of the $i^{th}$ random variable.

Another important concept of joint probability is the conditional probability density function. If information is known about one random variable, then one can utilize this information to find the conditional density function for two or more random variables. For example, for random variables $y_1$ and $y_2$ the condition distribution function is given as

$$f_{y_2|y_1}(y_2|y_1) = \frac{f(y_1, y_2)}{f_{Y_1}(y_1)} \tag{39}$$

which is the probability that $Y_2$ is equal to $y_2$ given that $Y_1$ is equal to $y_1$.

These basic joint probability concepts are used in the subsequent sections and are illustrated, for two random variables $x$ and $y$, in Figures 16 and 17. Figure 16 is a 3-D plot of the joint probability density function of the two variables where the sharp edge aligned with the diagonal of the variable plane illustrates strong positive correlation between the two variables. The marginal distributions of both variables are superimposed over contour areas of equal probability density in Figure 17. The direction of the diagonal dashed line passing through the center of the distribution in addition to the small amount of dispersion along the distribution contours visually illustrates the strong positive correlation of the two variables. Existing approaches for modelling joint randomness are now discussed.

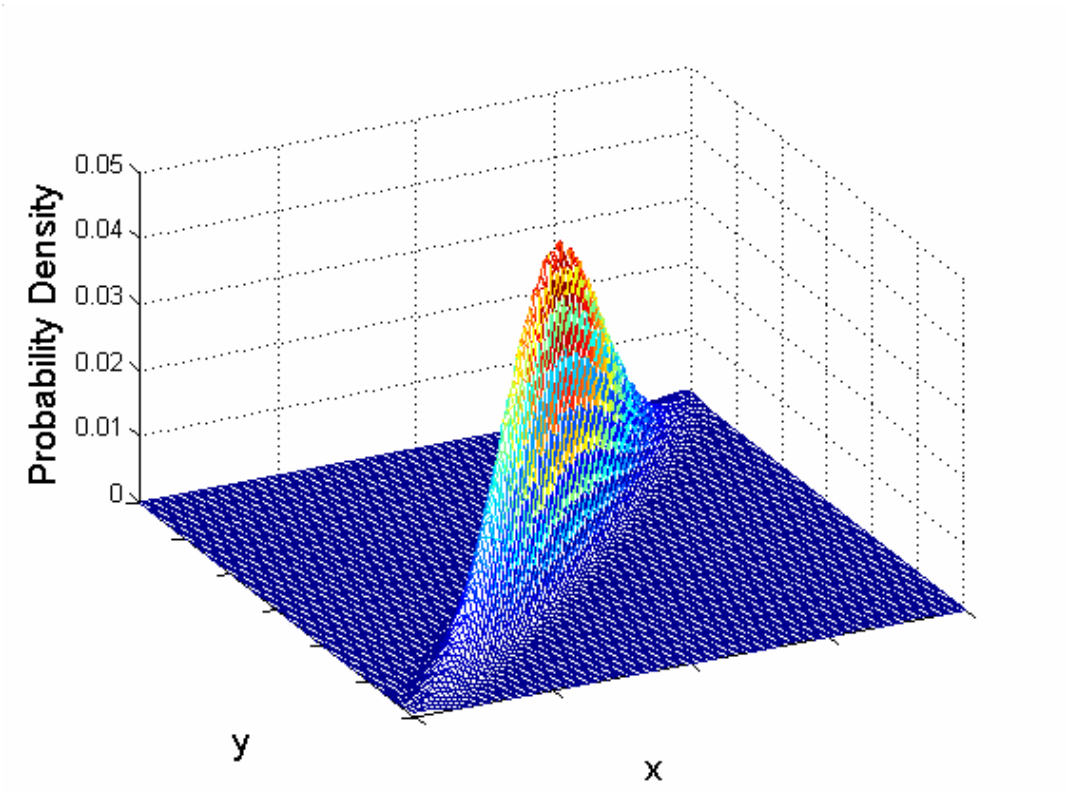


**Figure 16:** Bivariate Probability Space.

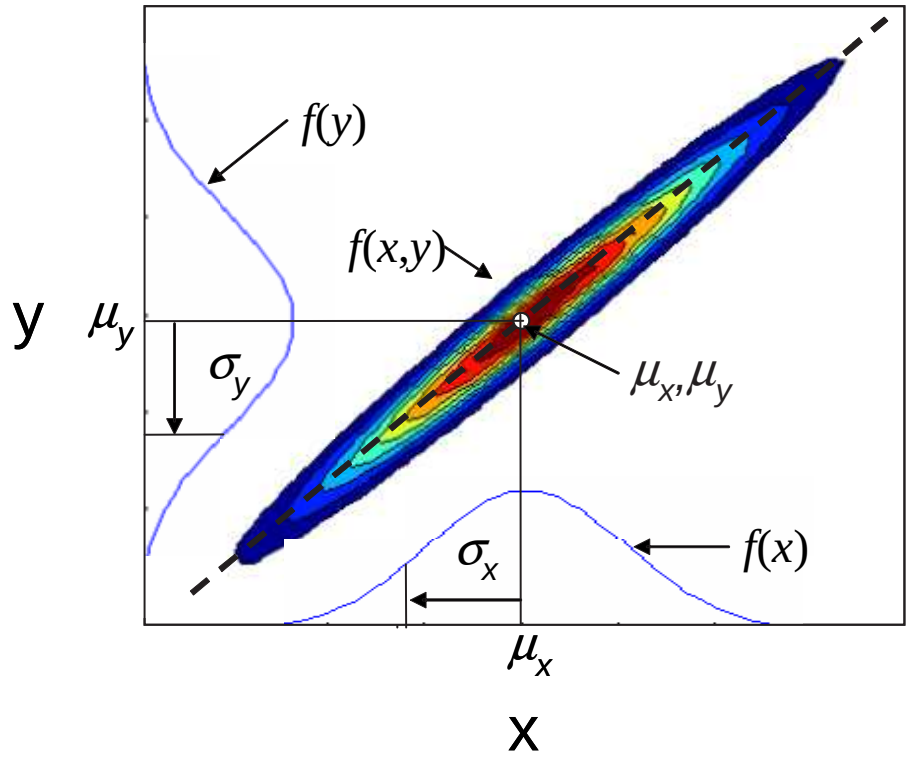


**Figure 17:** Multivariate Probability Terms.

## *3.3 Existing Joint Probability Models*

There have been numerous developments in the area of joint randomness modelling spanning several decades. However, only a handful of models have seen widespread acceptance in the field of structural reliability. These models as well as one that has recently been developed are now discussed.

### 3.3.1 Empirical Distribution Function

The most straightforward joint randomness model is the Empirical Distribution Function (EDF) [11]. As it is named, the model utilizes empirical data generated with either computational simulation, experimentation, or actual system measurements to create a joint randomness model based on the simulation counting technique. Available sample data, which could be generated from Monte Carlo simulation, is used

empirically to determine the joint multivariate distribution function

$$f(y_1, y_2, \ldots, y_m) = \frac{1}{N} \sum_{j=1}^{N} I(a_{j_1}, a_{j_2}, \ldots, a_{j_n} = y_1, y_2, \ldots, y_m) \tag{40}$$

where $N$ is the number of sample points, and $a_j$ are the sample point values corresponding to each variable. The indicator term $I(a_{j_1}, a_{j_2}, \ldots, a_{j_m} = y_1, y_2, \ldots, y_m)$ is simply a counting term and therefore is easily implemented within a computer program. This model, which is essentially a multi-response Monte Carlo approach, can be the most accurate approach given enough sample points. Within this study the EDF model is used as the datum method from which the results of other models are to be compared. The other methods included in this comparison are analytical in nature.

### 3.3.2 Constructive Models

There have been many attempts to develop a joint probability model using limited statistical information of the parameters. These models are called constructive models as they require, as input, a basic understanding of the parameter space, namely the marginal distributions and the covariance matrix of the random variables. The Morgensten model [67] was one of the first of this type to be implemented in structural reliability, however the range of the values of the correlation coefficient matrix proved to be severely limiting [55]. A more general approach is the Nataf joint Probability Density Function model (NPDF).

#### *3.3.2.1 Nataf Model*

NPDF requires both the marginal distribution and correlation matrix of the random variables under consideration. The joint randomness term is generated using a function of standard normal variates, $\boldsymbol{Z} = (Z_1, Z_2, \ldots, Z_m)$, which are determined through a marginal transformation of the random variables, $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_m)$,

given by

$$Z_i = \Phi^{-1}[F_{Y_i}(Y_i)], \text{ for } i = 1, \ldots, m \tag{41}$$

where $\Phi^{-1}(\cdot)$ is the inverse of the standard cumulative normal function. Assuming that $\boldsymbol{Z}$ is standard normal, then by the principle of inverse probability transformation the joint probability density function is given by

$$f_{Y_1,Y_2,\ldots,Y_m}(y_1, y_2, \ldots, y_m) = \underbrace{\frac{\varphi_m(\boldsymbol{z}, C')}{\varphi_1(z_1)\varphi_1(z_2)\cdots\varphi_1(z_m)}}_{\text{Joint Randomness}} \underbrace{\prod_{i=1}^{m} f_{Y_i}(y_i)}_{\text{Marginals}} \tag{42}$$

where $z_i = \Phi^{-1}[F_{Y_i}(y_i)]$, $\varphi_1(z_i)$ is the marginal standard normal PDF of the i$^{th}$ standard normal variate, $\varphi_m(\boldsymbol{z}, C')$ is the $m$-dimensional standard normal PDF of the standard normal variates, $\boldsymbol{Z}$, and $C'$ is the correlation matrix with elements $\rho'_{ij}$ determined through the following formula

$$\rho_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\frac{y_i - \mu_i}{\sigma_i}\right)\left(\frac{y_j - \mu_j}{\sigma_j}\right)\varphi_2(z_i, z_j, \rho'_{ij})dz_i dz_j \tag{43}$$

The solution of $\rho'_{ij}$ must be found iteratively through a numerical integration of equation 43. Der Kiureghian and Liu [29] have determined empirical formulas that provide $\rho'$ as a function of $\rho$ for several combinations of random variable distribution pairs. Because of the requirement that the correlation matrix $C'$ is positive definite, lower and upper bounds of $\rho$ must be adhered to [55], although for the Nataf model they are quite wide. For instance, in the case of a normal and uniform random variable pair, the upper and lower limits of $\rho$ are $-0.977$ and $0.977$, respectively. The NPDF model has become a popular method for creating a joint probability model of several random variables in addition to providing a joint standard normal transformation of an arbitrary set of random variables.

#### *3.3.2.2 Bandte Model*

A simpler approach was developed by Bandte [11] that, similar to the NPDF model in its general proposition, utilizes the marginal distributions as well as the covariance

matrix of the random variables. Bandte's proposed Probability Density Function model (BPDF) is given as

$$f_{Y_1,Y_2,\ldots,Y_m}(y_1,y_2,\ldots,y_m) = \underbrace{\frac{g(y_1,\ldots,y_m)}{C}}_{\text{Joint Randomness}} \underbrace{\prod_{i=1}^{m} f_{Y_i}(y_i)}_{\text{Marginals}} \tag{44}$$

where $C = \int_\Omega \ldots \int f_{y_1} \ldots f_{y_m} \cdot g(y_1,\ldots,y_n) dy_1 \ldots dy_m$. By inspection, the BPDF model has the same general elements as the NPDF model in that there is a term which is the product of the marginal distributions of the parameters and another term, $\frac{g}{C}$, which accounts for the joint random behavior between the variables. Due to the properties of the probability density function, the g-function must satisfy two conditions:

- $g \geqslant 0$ because of the non-negative criteria of the probability density function
- $g(y_1, y_2, \ldots, y_m) = 1$ when the random variables are independent.

Bandte, following the work of Tong [91] and Garvey [38], has proposed several explicit g-functions of two variables and one of a multivariate form. Each of these functions satisfy the constraints listed above. The multivariate g-function pursued by Bandte is given as

$$g(y_1,y_2,\ldots,y_m) = \binom{m}{2} + \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \rho_{y_i,y_j} \frac{y_i - y_{i,median}}{y_{i,range}} \frac{y_j - y_{j,median}}{y_{j,range}}$$

where $\binom{m}{2} = \frac{m!}{2!(m-2)!}$, $y_{i,median}$ is the median value of $y_i$ and $y_{i,range} = \frac{y_{i,max} - y_{i,min}}{2}$. This g-function, as with any explicit function, is appealing in the sense that it is easy to implement if the correlation coefficient matrix is known. However, it is only valid for the specified range and median values of the variables and thus any integration exercises with the model must be appropriately adjusted along these limits. As a result, the integral calculations can be affected by the choice of the ranges of the variables and so care must be taken to properly select these ranges.

The general BPDF method requires considerably less simulation points than the EDF method as the correlation matrix can be determined using a smaller set of data than completely describing the joint probability space using EDF. The proper g-function to use is up to the analyst to determine. At this time, only one multivariate g-function exists in the literature that meets the appropriate criteria. The BPDF method can, however, suffer from ill-behaved regions of the g-functions functions and can fail to model asymmetric joint distribution behavior [11]. Furthermore, these g-functions are strongly dependent on the choice of the variable ranges used in creating them. Consequently, the accuracy of the BPDF joint probability model is a strong function of the range of the variables used. A remedy to this caveat is to use as wide of ranges as possible for the given problem. This is particularly important when integral solutions are sought such as described in section 3.5 where one would use the joint probability model to compute the probability of an event occurring.

## *3.4 Sensitivity-Based Joint Probability Models*

The joint probability methods discussed thus far all require some knowledge of the response random variables, such as their marginal distributions and covariance matrix. But, in the case where the quantification of the joint randomness of variables local to a component reliability assessment is required, these methods would quickly become useless unless this information could be ascertained. Two new joint probability methods are now proposed that instead of relying on partial information of the response distributions, work directly with the response analysis(es) and the analysis input distributions to create the joint probability model of the responses. The first method is a multi-response extension of the venerable First Order Reliability Method (FORM), dubbed Multi-response First Order Second Moment (MFOSM) as it uses a sensitivity derivative of all $m$ responses and the input covariance to compute the $m$ by $m$ response covariance matrix and mean vector. Since it uses the sensitivity

derivative as input, it can be very efficient to implement. The second sensitivity-based model is the inverse transformation method, IPDF, discussed shortly, which also requires only a limited number of response evaluations, and can serve to quantify and characterize the joint randomness of such variables. An appealing characteristic of the solution produced using the inverse transformation method is that the solution is purely analytical and parametric with respect to the input variable statistics.

### 3.4.1 Multi-response First Order Second Moment

This sensitivity-based method involves approximating the gradient of the response vector, $\boldsymbol{y}$ , through a statistical based finite difference sampling of the response space. This is accomplished by implementing a partial derivative of the functional relationship between the each response and the vector of input variables, $y_i = g_i(\boldsymbol{x})$ where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. For instance, the sensitivity of the $i^{th}$ response variable, $y_i$, would be computed by solving for the partial derivative with respect to each input variable given by

$$\nabla y_i = \left\{ \begin{array}{c} \frac{\partial y_i}{\partial x_1} \\ \frac{\partial y_i}{\partial x_2} \\ \vdots \\ \frac{\partial y_i}{\partial x_n} \end{array} \right\} \tag{45}$$

However, in practice the functional relationship may be unknown. Thus, a numerical implementation of equation (45) would be required. The finite difference approach [9] is recommended for a deterministic sensitivity derivative calculation and would be of the following form

$$\frac{\partial y_i}{\partial x_j} \approx \frac{\triangle y_i}{\triangle x_j} \tag{46}$$

where $\triangle x_j$ would be chosen based on the type of numerical differentiation to be performed. For example, for forward finite difference the perturbation of each input variable would be a small scalar or adder applied to that input variable. If statistical information, such as mean and standard deviation, is available on the input variables,

one might use the Taylor Series Forward Finite Difference (TSFFD) estimation [9] procedure and conduct the sensitivity analysis about the mean value of the input vector using a range of plus one standard deviation about each input variable given as

$$\triangle \mathbf{x}_j = (\mu_{x_i} + \sigma_{x_i})$$

The remaining input variables in the vector would be set to their mean expected value. This calculation would be repeated for the perturbation of each input variable.

The baseline point at which the sensitivity measures are calculated should be taken at the expected value of the input vector, $\overline{\boldsymbol{x}}$, thus yielding the expected value of the response vector given as

$$E(\boldsymbol{y}) \approx g_i(\overline{\boldsymbol{x}}) \tag{47}$$

For the TSFFD method, the number of function evaluations required would be $n+1$ where $n$ is the number of input variables.

The covariance matrix between the responses can then be found using the partial derivative approximation from equation (45) as an input to the response covariance formula from Appendix A (section A.2). The sensitivity vector can be utilized to approximate the covariance matrix of the local input parameters as follows

$$COV(y_i, y_j) \approx \sum_{l=1}^{n} \sum_{m=1}^{n} \left(\frac{\triangle y_i}{\triangle x_l}\right) \left(\frac{\triangle y_j}{\triangle x_m}\right) COV(x_l, x_m) \tag{48}$$

where $n$ is the number of global input variables. Equation (48) reduces to the $VAR(y_i)$ when $COV(y_i, y_i)$ is sought and $\sigma_{y_i} = \sqrt{VARY(y_i)}$. Thus, the approximation of the correlation coefficient between the $i^{th}$ and $j^{th}$ response is given as

$$\rho_{y_i, y_j} = \frac{COV(y_i, y_j)}{\sigma_{y_i} \sigma_{y_j}} \tag{49}$$

As a final step in this method, assuming that the responses follow a joint normal distribution then the necessary mean and covariance parameters would be provided by the mean response vector (equation (47)) and response covariance matrix (equation

(48)) calculated using the TSFFD approach. The resulting joint normal distribution is given as

$$f_{\boldsymbol{y}}(\boldsymbol{y}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(\boldsymbol{y}-\overline{\boldsymbol{y}})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\overline{\boldsymbol{y}})\right] \tag{50}$$

where $T$ is the transpose operator and $\boldsymbol{\Sigma}$ is the covariance matrix whose $i,j$ element is $COV(y_i, y_j)$ [76]. Once the mean and covariance information is approximated using the statistical finite difference approach, the approximate joint normal probability density function can be found[3] The joint normal distribution is assumed in this method out of convenience. However, the sensitivity, or perturbation approach gleans no definitive information concerning the actual distribution of the joint probability space. This potential shortfall is addressed in sections 4.6.1.1 and 5.4.2.

### 3.4.2 Multi-response Inverse Transformation

The MFOSM method is appealing in that it can directly and efficiently approximate the response space statistics. But, it does not directly provide the joint distribution of the response space. An interesting principle in joint probability theory can be used to directly find the joint distribution of the response space and is called the inverse transformation principle which is essentially a mapping between probability spaces of the inputs and responses. The inverse transformation principle is useful when one seeks the joint probability of several variables which are the responses of a system of input variables. The inverse method directly accounts for both input variable dependency and non-normalcy and should necessary assumptions be valid for the system of equations, an exact solution for the joint probability of the responses can be found. Consider one of the analyses which produces several outputs. Each analysis requires numerous inputs including potentially those from other analyses. The response or output parameters are labelled as the functions, $y_1, y_2, \ldots, y_m$, and

[3]This approach can be thought of as a multi-response extension of Wu's single response Mean Value (MV) method [99]. The multi-response capability for the MFOSM is enabled using the covariance approximation formula of Appendix A.

their inputs defined as $x_1, x_2, \ldots, x_n$. Thus the functions can be represented as

$$\begin{aligned} y_1 &= g_1(x_1, \ldots, x_n) \\ &\vdots \\ y_m &= g_m(x_1, \ldots, x_n) \end{aligned} \tag{51}$$

There are two conditions must be satisfied before using the inverse method:

- The inverse of the solution of system of equations must exist and be unique and
- The response functions must have continuous partial derivatives at all values of the input variables (i.e. the Jacobian is non-zero at all points).

Assuming that the inverse transformation is unique, then the input variables can then be expressed in terms of the responses in the following way

$$\begin{aligned} x_1 &= h_1(y_1, \ldots, y_m) \\ &\vdots \\ x_n &= h_n(y_1, \ldots, y_m) \end{aligned} \tag{52}$$

And given that the joint probability distribution function, JPDF, of the input variables is known, then the JPDF of the response functions is

$$f_{Y_1,\ldots,Y_m}(y_1, \ldots, y_m) = f_{X_1 \ldots X_n}(x_1, \ldots, x_n)|J(x_1, \ldots, x_n)|^{-1} \tag{53}$$

where $J$ which is the Jacobian of the transformation. Considering the popularity of linear statistical models such as the response surface method and power series approximations (i.e. Taylor series) in approximating complex implicit analysis routines, the inverse method although not a relatively new development can work directly with these approximations to determine the joint probability distribution of multiple responses between different blocks in a multidisciplinary analysis environment.

*An example is now given:*

Consider two functions of three independent normal random variables given as

$$y_1 = M_{11}x_1 + M_{12}x_2 + M_{13}x_3$$
$$y_2 = M_{21}x_1 + M_{22}x_2 + M_{23}x_3$$

Since the inverse method requires symmetry of the coefficient matrix for inversion, we then define an auxiliary function

$$y_3 = M_{31}x_1 + M_{32}x_2 + M_{33}x_3$$

The third equation is important for the inversion of the problem yet will not affect the final solution, as will be shown, since the JPDF of $y_1$ and $y_2$ is sought. The system of equations can then be expressed as $\mathbf{y} = M\mathbf{x}$ where $M$ is the matrix of coefficients. The inverse transformation is therefore $\mathbf{x} = M^{-1}\mathbf{y}$.

Now consider a simple case where $M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$. Note that each row of M can be thought of as a vector of regression coefficients from a response surface equation or power series approximation. The inverse of $M$ is $M^{-1} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -2 \end{bmatrix}$, which results in the determinant of the Jacobian equal to 1 and an inverse transformation of the system of equations given as

$$\begin{aligned} x_1 &= y_3 \\ x_2 &= -y_2 + y_3 \\ x_3 &= y_1 + y_2 - 2y_3 \end{aligned}$$

Using equation (53) we can then express the joint distribution of the 3 response functions as

$$f_{Y_1,Y_2,Y_3}(y_1, y_2, y_3) = f_{X_1,X_2,X_3}\left(\{y_3\}, \{-y_2 + y_3\}, \{y_1 + y_2 - 2y_3\}\right)$$

Since the input variables are independent normal random variables, their joint probability distribution function reduces to

$$f_{X_1,X_2,X_3}(x_1,x_2,x_3) = f_{X_1}(x_1)\cdot f_{X_2}(x_2)\cdot f_{X_3}(x_3) = \prod_{i=1}^{3}\frac{1}{\sigma_{x_i}\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x_i-\mu_{x_i}}{\sigma_{x_i}}\right)^2} \quad (54)$$

Substituting equation (54) into equation (53) and substituting variables allows us to solve for the joint probability distribution function of the three responses given as

$$f_{Y_1,Y_2,Y_3}(y_1,y_2,y_3) = \frac{1}{\left(\prod_{i=1}^{3}\sigma_{x_i}\right)(2\pi)^{\frac{3}{2}}}e^{-\frac{1}{2}\left[\left(\frac{y_3-\mu_{x_1}}{\sigma_{x_1}}\right)^2+\left(\frac{y_3-y_2-\mu_{x_2}}{\sigma_{x_2}}\right)^2+\left(\frac{y_1+y_2-2y_3-\mu_{x_3}}{\sigma_{x_3}}\right)^2\right]} \quad (55)$$

The JPDF of $y_1$ and $y_2$ can now be solved by integrating equation (55) with respect to $y_3$ over the span of real numbers. This function clearly works well with linear approximations and accounts for the individual distributions of each input variable.

Of particular interest to the goals of this study is how to model the effect of changes in the system and component input variables on the probabilistic response distributions. An interesting feature of the IPDF joint probability model is that this functional relationship is implicit to the resulting solution. Stated another way, the IPDF solution directly accounts for any changes in the input variable properties such as the mean value or variance vectors. In this respect, the IPDF method is superior to the alternative methods considered in this study.

### 3.4.3 Joint Covariate Model

Interestingly, the JPDF methods that are based on the product of the individual marginal probability distribution functions, such as BPDF and NPDF, can be modified to have a similar parametric capability as the IPDF method. The modification involves using the covariate method discussed in section (2.4.2) by replacing the marginal distribution functions of the JPDF with parametric covariate models. Assuming that the correlation coefficient matrix remains constant over the space of

covariate values under consideration, the modified models would therefore provide a unique and straightforward means of modeling both the parametric response distribution behavior as well as joint randomness between the responses.

Using the BPDF model, the resulting generalized covariate joint probability function is obtained by combining equations (19) and (44) as follows

$$f_{\mathbf{Y}}(\mathbf{Y},\mathbf{x}) = \frac{g(y_1, y_2, \ldots, y_n)}{C} \prod_{i=1}^{n} f_{Y_i}(y_i, \mathbf{x}) \tag{56}$$

where $f_{Y_i}(y_i, \mathbf{x}) = -S^{'}(y_i, \mathbf{x})$. Bandte's joint probabilistic method is straightforward to implement, but it can be limiting depending on the g-function selected.

The generalized covariate joint probability function using the NPDF model would take the following form when covariate models are used

$$f_{\boldsymbol{Y}}(\boldsymbol{Y},\boldsymbol{x}) = \frac{\varphi_n(\boldsymbol{z}, C^{'})}{\varphi_1 \varphi_2 \cdots \varphi_n} \prod_{i=1}^{n} f_{Y_i}(y_i, \boldsymbol{x}) \tag{57}$$

where $Z_i(\boldsymbol{x}) = \Phi^{-1}[F_{Y_i}(Y_i, \boldsymbol{x})]$ and is the normal transformation of the covariate marginal distribution of the $Y_i^{th}$ response. It is assumed that the correlation coefficient matrix, $C^{'}$, is independent of the covariate vector, $\boldsymbol{x}$. However, joint randomness can be highly dependent on the covariate variables and thus the correlation matrix should be recalculated at the various covariate variable values tested to verify the required independence. Should the joint covariate NPDF model be applicable, integration over the joint standard normal space produced by the Nataf transformation can be accomplished using an approach developed by Gollwitzer and Rackwitz [42].

Interestingly, the IPDF model and a joint covariate model can be combined to increase the flexibility of creating a parametric joint probability model. For instance, if the marginal distributions and correlation coefficient of two variables, $y_1$ and $y_2$, were already known and a system of equations were provided for the additional variables, $y_3$ and $y_4$, then a combined joint probability model might take the following form

$$f_{\boldsymbol{Y}}(\boldsymbol{Y},\boldsymbol{x}) = \underbrace{f_{Y_1,Y_2}(y_1, y_2, x_1, x_2)}_{\text{Joint covariate model}} \cdot \underbrace{\frac{1}{|J|} f_{X_3,X_4}(x_3, x_4)}_{\text{IPDF model}}$$

where $\boldsymbol{x}$ is the input vector, $x_1$ and $x_2$ are covariate variables, and $x_3$ and $x_4$ are the solutions of the inverse of the system of equations which would be expressed as a function of $y_3$ and $y_4$. Note that this formulation assumes that the two pairs of variables, $y_1$ & $y_2$ and $y_3$ & $y_4$, are independent.

## *3.5 Event Probability Calculation*

Creating a joint probability model of a set of variables, as demonstrated using both constructive and sensitivity-based methods, is only one step in a full probabilistic analysis. In the context of this study, there are two applications of the JPDF model. The first is a direct probability analysis by appropriately integrating the JPDF over an integral space for probability solutions. The second application is the use of the JPDF model as input to a subsequent probabilistic analysis which is the approach taken within this study (Chapter 4) to analyze a component operating in a complex system. In this application, the upstream analyses would be queried to generate a JPDF of the variables that are input for the component reliability analysis.

### 3.5.1 Probability Solution of Joint Probability Model

In any case, all of the methods discussed in this chapter share the same purpose; to provide a joint probability distribution of the random variables under consideration. To calculate the probability of an event occurring within this response space, the joint probability density function must be integrated. Two integral forms are now given, representing two different failure event categories: weakest link (series-OR), or redundant system (parallel-AND). For a series event scenario of dependent component lifetimes the cumulative distribution function[4] modified to include the vector of

[4]The original formula is from Leemis [53] but is generalized here for joint covariate models.

covariates, $\boldsymbol{x}$, is given as

$$F_{\boldsymbol{Y}}(\boldsymbol{y},\boldsymbol{x}) = \sum_i F_{Y_i}(y_i,\boldsymbol{x}) - \sum_{i<j} F_{Y_i\cap Y_j}(y_i,y_j,\boldsymbol{x}) + \sum_{i<j<k} F_{Y_i\cap Y_j\cap Y_k}(y_i,y_j,y_k,\boldsymbol{x}) - \ldots + (-1)^{m+1} F_{Y_1\cap Y_2\cap\ldots\cap Y_m}(\boldsymbol{y},\boldsymbol{x}) \tag{58}$$

where $F_{Y_1\cap Y_2\cap\ldots\cap Y_m}(\boldsymbol{y},\boldsymbol{x}) = \int_{-\infty}^{y_n}\ldots\int_{-\infty}^{y_1} f(\boldsymbol{y},\boldsymbol{x})dy_1dy_2\ldots dy_m$ which is the overlapping event space for all $m$ responses.

For a parallel event system the cumulative distribution function modified to include the vector of covariates is given as

$$\begin{gathered} F_{\boldsymbol{Y}}(\boldsymbol{Y},\boldsymbol{x}) = 1 - \int_{y_m}^{\infty}\ldots\int_{y_1}^{\infty} f(\boldsymbol{y},\boldsymbol{x})dy_1dy_2\ldots dy_m \\ or \\ F_{\boldsymbol{Y}}(\boldsymbol{Y},\boldsymbol{x}) = \int_{-\infty}^{y_m}\ldots\int_{-\infty}^{y_1} f(\boldsymbol{y},\boldsymbol{x})dy_1dy_2\ldots dy_m \end{gathered} \tag{59}$$

Choosing which form of equation (59) to use depends on the ease of integration.

Equation (58) would be used when the event criteria is the union of the response events while equation (59) is used when the event criteria is the intersection of the various response events. For example, if one were interested in finding the probability that response one OR response two are less than a certain value, then this would become a union event problem and thus require solving equation (58). Otherwise, if the probability statement was the chance that response one AND response two would be less than or equal to a certain value, then equation (59) would be used.

The solution of these integrals can at times be difficult to find. Although many can be solved using symbolic mathematics, they can easily be of a form that must be solved using numerical integration. Often, the direct integration of the joint density function is possible. Otherwise, one might transform the model into one that can be more readily integrated. The integration of the JPDF equations is necessary for the validation and evaluation discussed in section 3.6 and is done so using symbolic as well as numerical integration.

### 3.5.2 Transformation of Probability Space

Many probabilistic analysis methods such as FORM and SORM assume that the random input variables are uncorrelated and normally distributed. If this assumption is not valid, then the vector of basic random variables must be transformed into an equivalent vector of independent standard normal variables.

For the case when the marginal Normal distributions and covariance matrix are known, the following eigenvalue and Cholesky factorization methods can be used. In either method the input variables can be converted to a reduced form as follows

$$x_i^{'} = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}}, \ (i = 1, 2, \ldots, n) \tag{60}$$

It can be shown that the covariance matrix of the reduced variables takes the following form

$$[C'] = \begin{bmatrix} 1 & \rho_{x_1,x_2} & \cdots & \rho_{x_1,x_n} \\ \rho_{x_2,x_1} & 1 & \cdots & \rho_{x_2,x_n} \\ \vdots & \vdots & 1 & \vdots \\ \rho_{x_n,x_1} & \rho_{x_n,x_2} & \cdots & 1 \end{bmatrix} \tag{61}$$

where $\rho_{x_i,x_j}$ is the correlation coefficient of the i$^{th}$ and j$^{th}$ variables. Using the eigenvalue method [45], the equivalent uncorrelated variables can be solved using

$$\{X\} = \left[\sigma_X^N\right][T]\{Z\} + \left\{\mu_X^N\right\} \tag{62}$$

where T is the orthogonal transformation matrix of the eigenvectors of the correlation matrix $[C']$. The Cholesky factorization method solves for the equivalent transformed variates using the following formula

$$Z = L^{-1}(X^{'})^t \tag{63}$$

where L is the lower triangular matrix found using Cholesky factorization. The Cholesky factorization method, equation (63), is recommended over the approach of equation (62) when there is a large number of random variables. Both variants of

the uncorrelation transformation given by equations (63) or (62) are exact transformations.

Interestingly, for the case where the marginal distributions are non-normal, the Nataf method can be used to provide an approximate transformed joint standard-normal space which usually is quite accurate. This space can be further transformed to an independent standard normal space using Cholesky factorization discussed earlier. This process of creating an independent, multinormal probability space is a necessary step to implement the most probable point methods (FORM and SORM) discussed in section 2.3.

When the joint cumulative distribution of all the variables is known, a unique and equivalent set of independent normal variables can be found using Rosenblatt's transformation [77]. This transformation uses a successive conditioning of the dependent variables given as

$$\begin{aligned} \Phi(u_1) &= F_X(x_1) \\ \Phi(u_2) &= F_{X_2|X_1}(x_2|x_1) \\ &\vdots \\ \Phi(u_n) &= F_{X_n|X_1,X_2,\ldots,X_{n-1}}(x_n|x_1,x_2,\ldots,x_{n-1}) \end{aligned} \tag{64}$$

This formulation is a natural representation of a stochastic process but is rarely applicable in the previous structural reliability literature outside of stochastic process applications. Such a condition where the joint cumulative distribution function is known is rare and, more often than not, difficult to determine directly.

Once these transformations are performed, an equivalent independent standard normal space is defined. For the case where a jointly distributed normal space is uncorrelated, the transformation is an exact solution requiring no approximation of the pre-transformed space. The Nataf method is an approximation of the pre-transformed space which should be considered when transforming spaces of an irregular type.

Since the resulting transformed space is independent and standard normal, the joint density function then becomes

$$f_U(u) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} exp\left(\frac{-u_i^2}{2}\right) du \tag{65}$$

where $\boldsymbol{u}$ is the vector of standard normal variates. The form of equation (65) is readily used in many reliability analysis methods such as FORM and SORM.

## *3.6 Numerical Validation and Evaluation*

Several methods for creating joint probability models have been described in this chapter. However, a thorough evaluation and, in the case of the MFOSM and IPDF methods, validation is necessary. To maintain the ease of replicating the process pursued herein for validating and evaluating the various methods, simple cases are used. Two cases, one with jointly distributed normal input and one with independent non-normal input, for a linear system of equations are considered. Then, a quadratic system of equations with joint normal input is considered. In either of these three cases, a JPDF is created using all of the joint probability models described in this chapter. The EDF method is used in each case as the baseline solution from which the alternative methods, MFOSM, IPDF, BPDF, and NPDF, are to be compared. Graphical comparisons of the JPDF models are made and probability solutions calculated using each of the JPDF models for a specified bivariate failure condition.

### 3.6.1 Linear Responses with Joint Normal Input

The first case considered is for a linear system of equations with joint normal input. The problem statement is to compute the reliability of a component under two failure mechanisms defined as $y_1 = x_1 + x_2$ and $y_2 = x_1 + 2x_2$ where $x_1$ and $x_2$ are jointly distributed normal random variables with $\rho = 0.5$, $\mu = 20$, $\sigma = 2$. Failure occurs when either $y_1 \geqslant 46.5$ or $y_2 \geqslant 70$.

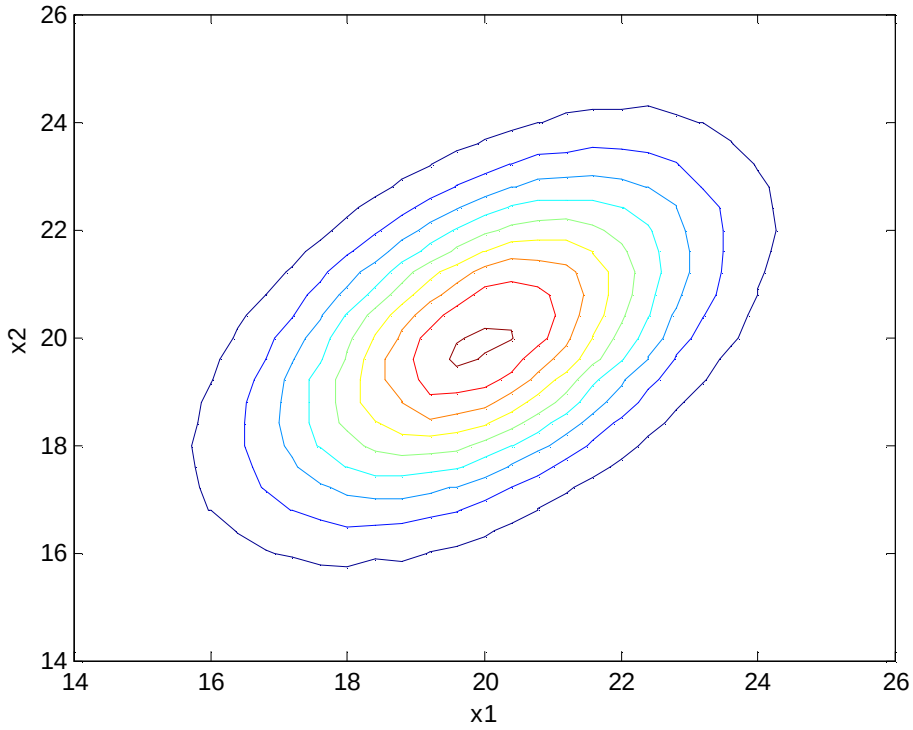


**Figure 18:** Joint Normal Input Space ($\rho = 0.5$).

The baseline solution using EDF is sought first. The implementation of the EDF method is rather straightforward for this problem. It entails conducting a Monte Carlo analysis that creates a data set of response values for $y_1$ and $y_2$ as a function of the joint normal input variables $x_1$ and $x_2$. The bivariate joint normal input distribution is modeled using the following sampling equation [76]

$$\begin{aligned} x_1 &= \mu + \sigma U \\ x_2 &= \mu + \sigma\rho U + \sigma V\sqrt{(1-p^2)} \end{aligned}$$

where $U$ and $V$ are independent standard normal variates. The correlation coefficient, $\rho$, can be changed to zero in this formula to model independence of the input variables. Figures 18, 19, and 20 show the input and response joint distribution functions, respectively, for the jointly distributed input variables ($\rho = 0.5$). Notice the moderate covariance of the input and even stronger covariance of the response as exhibited by

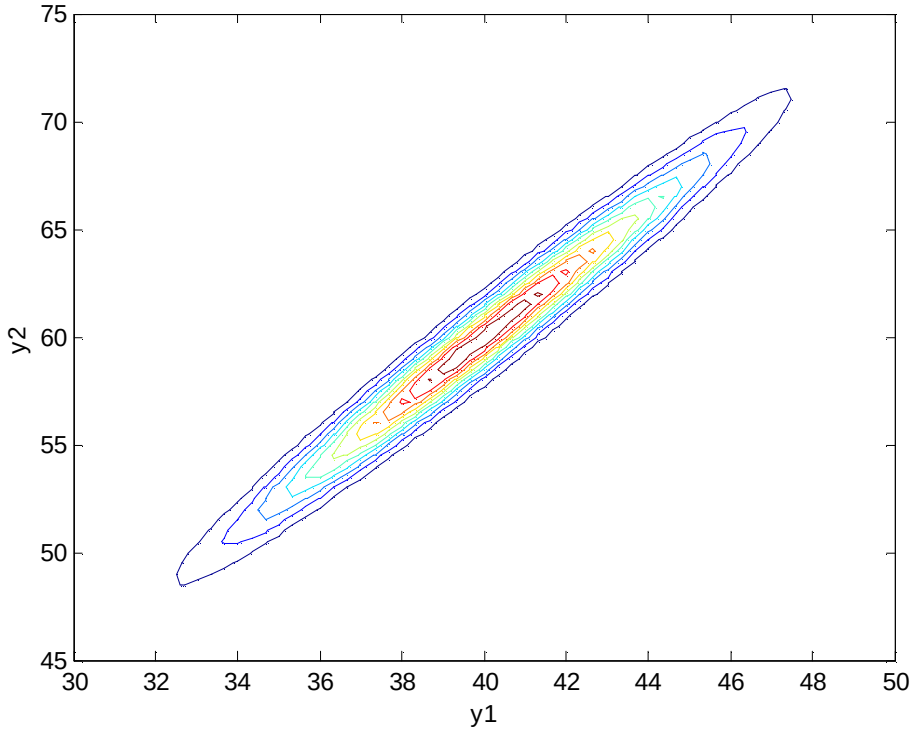


**Figure 19:** Linear System Joint Probability Solution (Joint Normal Input).

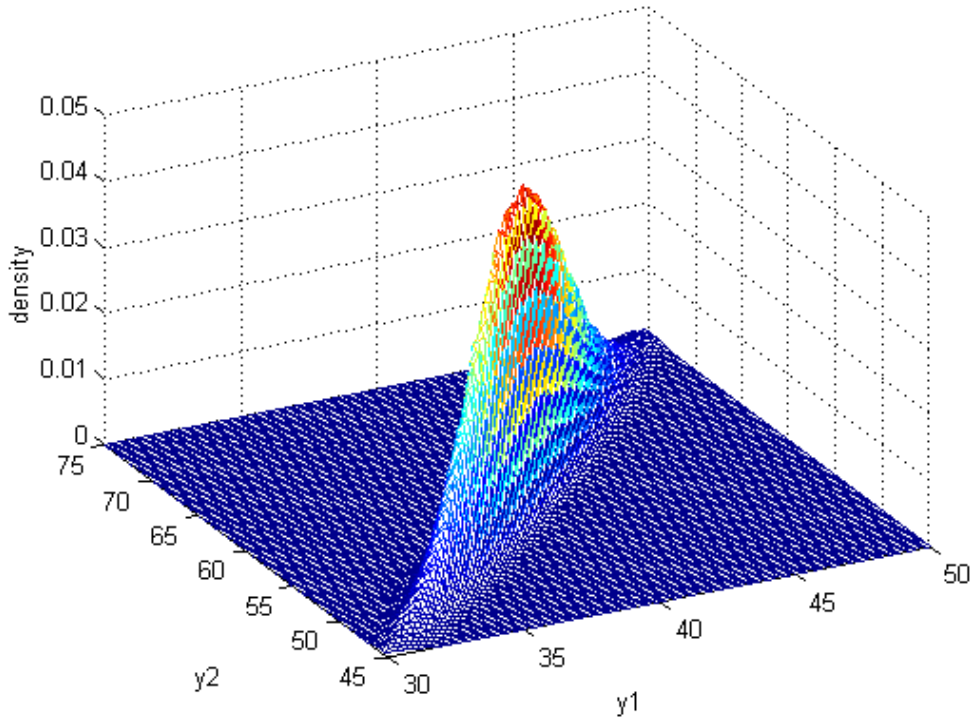


**Figure 20:** Linear System Joint Probability Solution (Joint Normal Input).

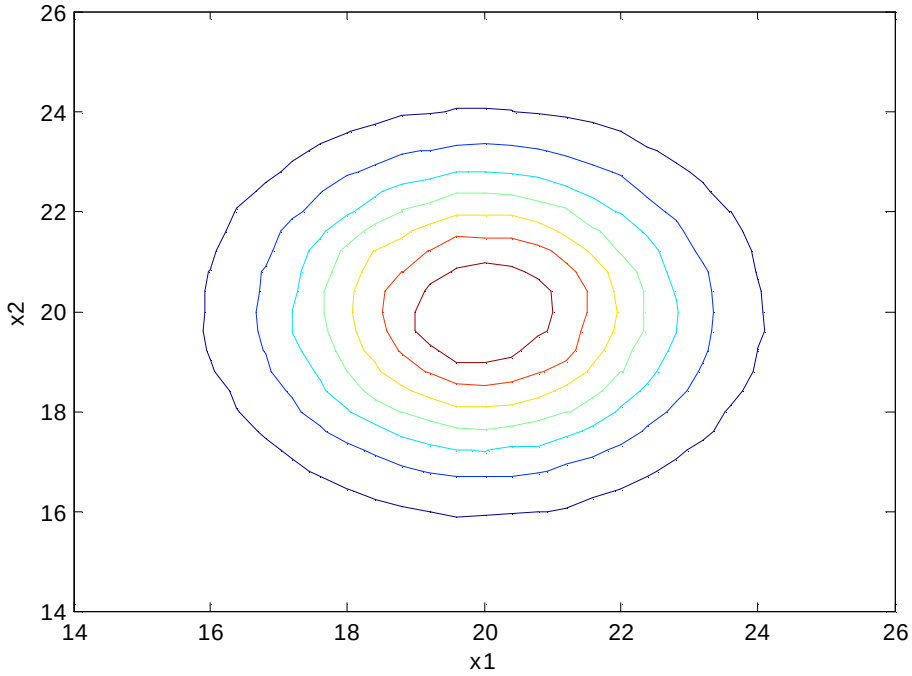


**Figure 21:** Independent Normal Input Space ($\rho = 0$).

the skewness of the distributions. If the input variables were independent, then the input distribution would become spherically symmetric and the response distribution would exhibit less covariance as shown by Figures 21 and 22, respectively. However, notice that even with independent input the response distribution exhibits considerable joint randomness due to the functional relationships between each response and the input variables.

Finally, to calculate the probability of failure given the failure definition a simple counting technique was implemented to determine the number of failed cases. An initial simulation size of $10,000$ cases was selected and increased all the way up to $1,000,000$ simulations to achieve a stabilized and accurate probability of failure of 0.03514 for the correlated input variables. As a comparison, the probability of failure when assuming the responses and input are independent were also calculated and are reported in Table 2. By failing to model the correlation of the input variables,

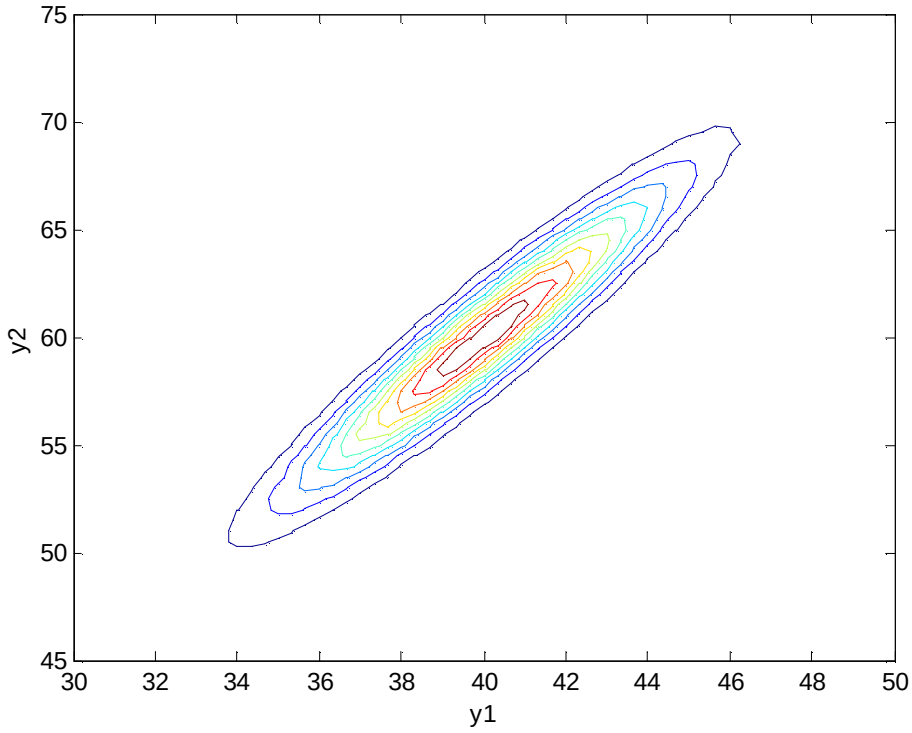


**Figure 22:** Linear System Joint Probability Solution (Independent Input).

**Table 2:** Linear System Failure Probability Results.

| | | Method | | | | | |
|---|---|---|---|---|---|---|---|
| **Output** | **Input** | **EDF** | **FPI** | **BPDF** | **MFOSM** | **NPDF** | **IPDF** |
| Ind. | Ind. | 0.02325 | 0.02332 | N/A | N/A | N/A | N/A |
| | Corr. | 0.05878 | 0.05880 | N/A | N/A | N/A | N/A |
| Joint | Ind. | 0.01570 | N/A | 0.02330 | 0.01562 | 0.01562 | 0.01564 |
| | Corr. | 0.03514 | N/A | 0.06098 | 0.03497 | 0.03463 | 0.03497 |
| Evaluations | | 1E6 | 3 | 1E6 | 5 | 1E6 | 5 |

the resulting probability of failure is under predicted by a factor of 2, and thus is non-conservative. The opposite is true for considering the joint randomness of the responses. For this condition, the calculated probability of failure is increased by a factor of around 1.7 when modelling the responses as independent, and thus, although less accurate, is more conservative. The response distribution is more concentrated about the mean points when joint randomness is considered, thus reducing the amount of volume of the distribution outside of the failure boundary. This explains why the probability of failure is less when considering joint randomness. Another result, albeit intuitive, of the EDF model implementation is that considerable dependency exists between the two responses as the correlation coefficient was computed to be 0.98. A correlated response distribution is to be expected for component failure modes since they usually are functions of the same variables, i.e. stress, geometry, or material properties.

The modern approach to computing the probability if failure is to use one of several methods available under the category of Fast Probability Integration (FPI) methods which are essentially the advanced structural reliability methods described in section 2.3 and summarized by Wu [101]. In either of the validation problems the most accurate FPI method result, as compared to the EDF solution, was chosen as the reported FPI result. For the linear system of joint normal input, the results of the FPI method results are also given in Table 2. The FPI methods are very accurate for this problem when considering either correlated or uncorrelated input variables. However, they are limited to the case where the responses are assumed to be independent and can not capture the joint randomness of the response, which for this problem should not be neglected.

To implement the MPDF method, a joint normal probability density function (equation (50)) is to be created using the covariance approximation technique derived in Appendix B to estimate the multivariate covariance matrix between the responses

(equation (48)). The Taylor series finite difference technique is employed to construct the first order approximation of the gradient for both responses which is an input to the covariance approximation technique. Interestingly, for this problem the covariance matrix solution was found to be

$$COV\left[y_1, y_2\right] = \begin{bmatrix} 12 & 18 \\ 18 & 28 \end{bmatrix}$$

and $\rho_{y_1,y_2} = 0.98$ which is the same solution produced using the accurate EDF method. But, this is to be expected since the bivariate probability density space of a linear system of equations with joint normal input is also joint normal and so the approximation technique for this situation should be exact. Substituting this result into the joint normal probability density function (equation (50)) allows the computation of the failure probability as 0.03497 for joint normal input ($\rho = 0.5$) and 0.01562 for the case where the input is independent normal. Comparing these results to those found using the EDF method one can conclude that a joint normal response space of a linear system of equations can be accurately approximated using the MPDF method.

The BPDF method is rather straightforward to implement for this problem since the EDF data set is available. The usual case, as that pursued by Bandte [11], is where the marginal distributions of the parameters are known but their mutual correlation is unknown. However, for this simple example both the marginal distributions and correlation coefficient matrix can be determined with minimal effort using the EDF data. The correlation coefficient matrix is found to be 0.98 between the two responses while both were normally distributed.

To create the BPDF model, the joint randomness is modelled through the use of a g-function which is multiplied with the product of the marginal distributions (reference equation (44)). Bandte explored several potential g-functions that met the aforementioned requirements. The multivariate (i.e. three or more variables)

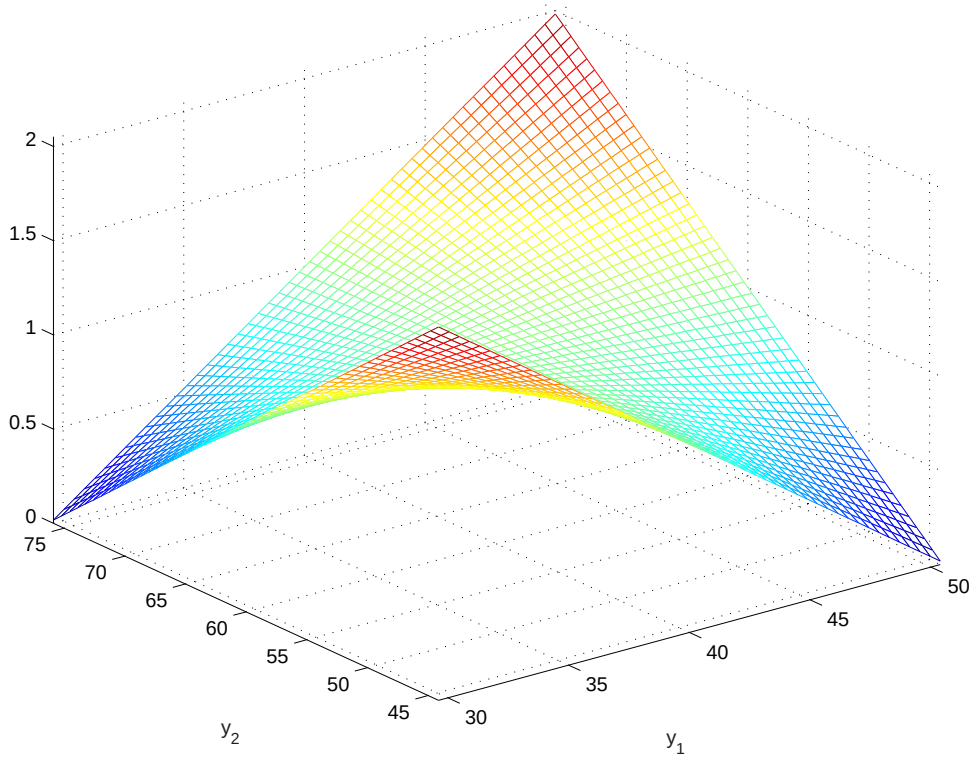


**Figure 23:** BPDF *g-function* ($\rho = 0.98$).

g-function explored by Bandte [11] for two variables is given as

$$g(y_1, y_2) = 1 + \rho \frac{y_1 - y_{1,median}}{y_{1,range}} \frac{y_2 - y_{2,median}}{y_{2,range}}$$

The response variable range input for this g-function was provided by defining the maximum and minimum values of each variable to be equal to the mean plus or minus three standard deviations of that variable. The resulting g-function is shown in Figure 23. The general form of this simple g-function always results in a positive value for the g-function as long as the variable values are within their maximum and minimum interval. This interval requirement creates a constraint on the limits of integration when evaluating the joint probability model. This constraint should be carefully evaluated as a violation of this could result in negative density distribution values or possibly an inferior integration space when computing cumulative distribution values. Therefore, it is recommended that the range of the variables be quite large (i.e. at

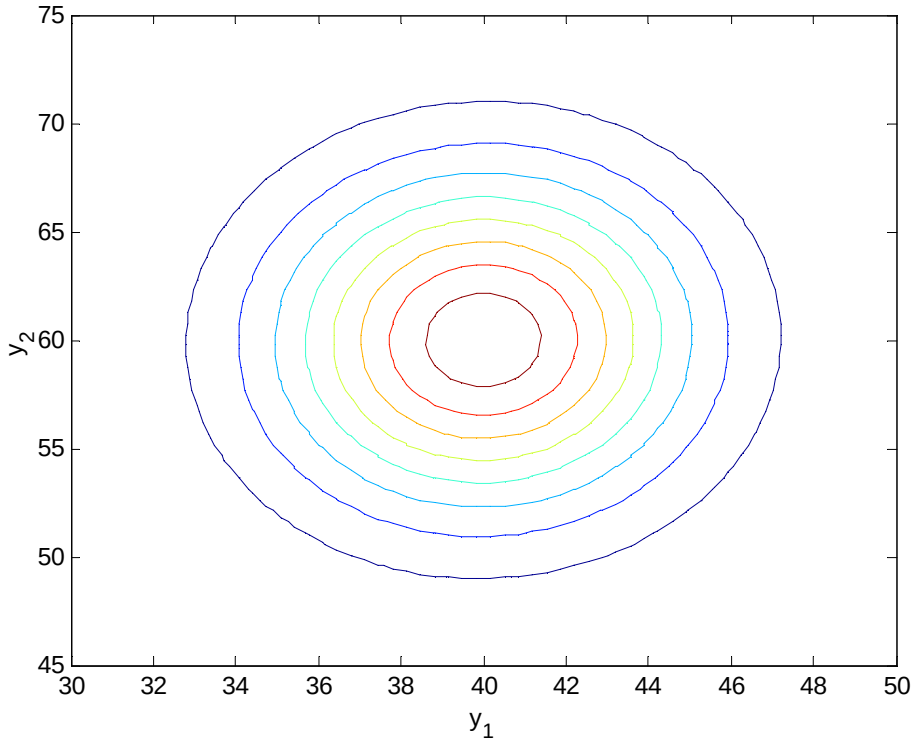


**Figure 24:** Linear System Joint Distribution Using BPDF.

least $+/-3\sigma$). The probability of failure is determined analytically to be 0.06098 by integrating the joint distribution created by the BPDF model and is reported for independent and correlated input in Table 2. This solution is surprisingly close to the solution obtained by assuming the responses are independent. An inspection of the joint probability response predicted by the BPDF method, shown in Figure 24, illustrates that this implementation of the BPDF method does not capture the joint randomness between the two responses. Little or no skewness is found in the joint probability response and therefore the two random variables are graphically depicted as independent. This would explain why the BPDF method produces a probability calculation result similar to the EDF solution when the responses are assumed to be independent. As the proposition by Bandte is theoretically sounds, the limiting component of this implementation must be the multivariate g-function chosen. This is discussed further as each additional method is implemented.

Next, the Nataf model is implemented to create an analytical JPDF. The first step of the NPDF method is to solve for $\rho^{'}$. By the theory of linear functions of Gaussian (Normal) distributions, the joint probability space of the two responses is a bivariate normal distribution. So the marginal distribution of both responses is normal and the correlation coefficient is equal to 0.98. Since the two responses form a joint normal probability density space, $\rho^{'} = \rho = 0.98$ and the NPDF method can be implemented with ease. The solution using the NPDF model must be found using a numerical integration scheme and is also reported in Table 2. The solution is rather accurate as the result is just over 1% different from the baseline EDF solution. For this problem the NPDF model is much more accurate than the BPDF model and is just as easy to implement since the approximate correlation coefficient for the transformed space, $\rho^{'}$, did not have to be solved for. The NPDF model graphically appears to successfully capture the joint randomness of the system, as shown in Figures 25 and 26, by an almost perfect replica of the joint probability solution. Furthermore, the g-function produced using the NPDF model, as shown in Figure 27, is very different from the g-function used in the BPDF method, shown in Figure 23. A small ridge is present along the linear dependence line and asymptotically increases much farther away from the JPDF center of mass. Thus, the JPDF modelled is weighted strongly in the area far from the center of mass which explains the non-concentric circles of equal probability shown in Figure 25.

The inverse transformation model (IPDF) is applied as follows. The inverse solution of the joint probability distribution function of the two responses is found using equation (53) repeated here

$$f_{Y_1,Y_2}(y_1, y_2) = |J|^{-1} f_{X_1,X_2}(x_1, x_2)$$

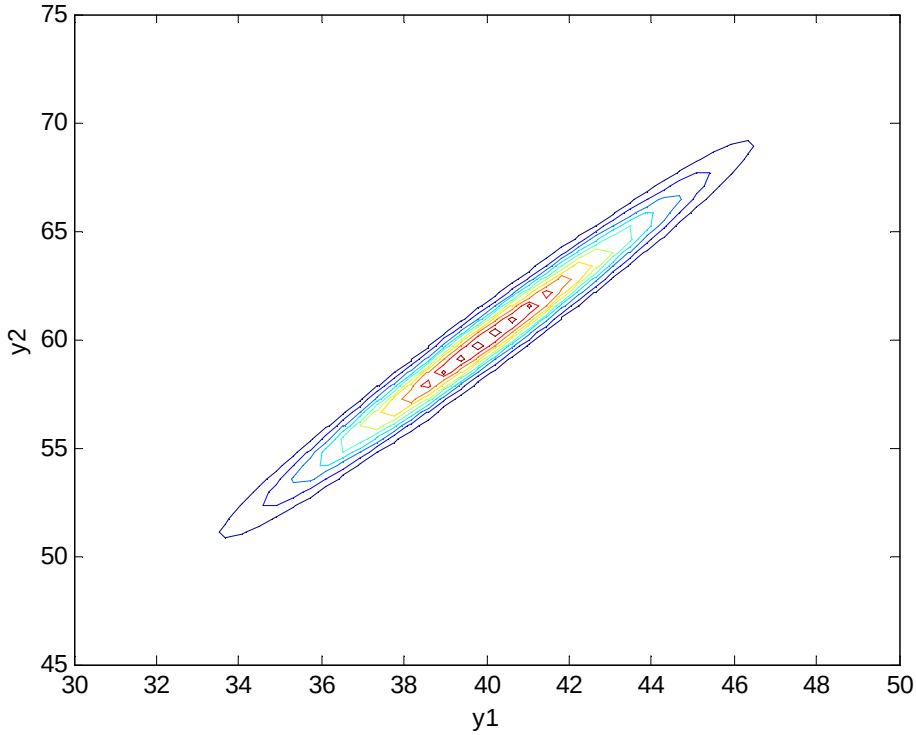


**Figure 25:** Linear System Joint Distribution Using NPDF.

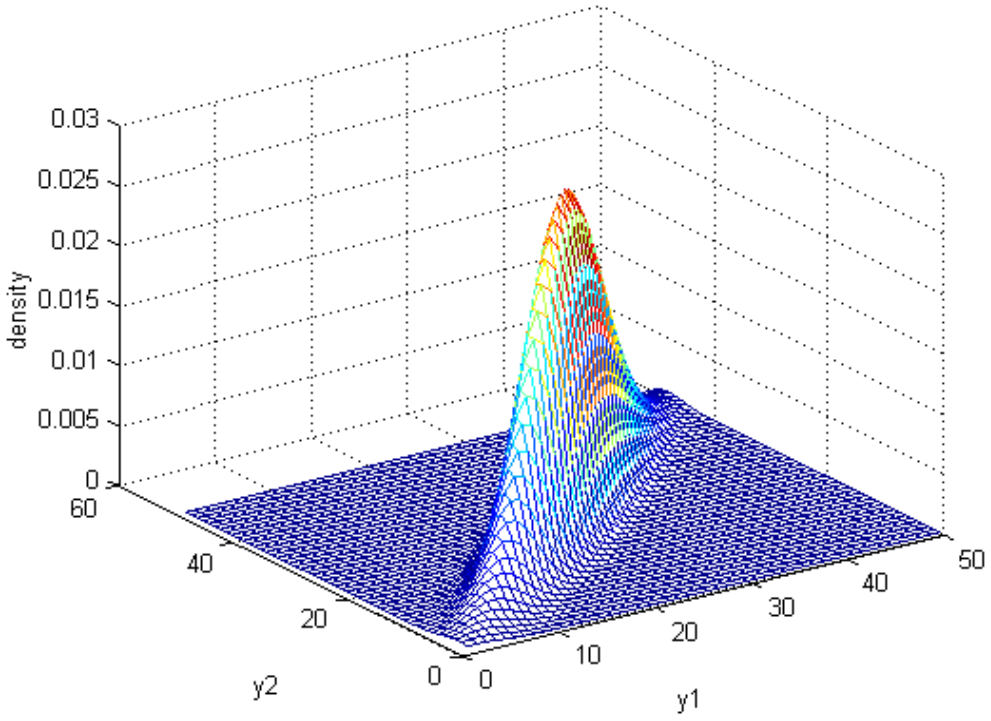


**Figure 26:** Linear System Joint Distribution Using NPDF.

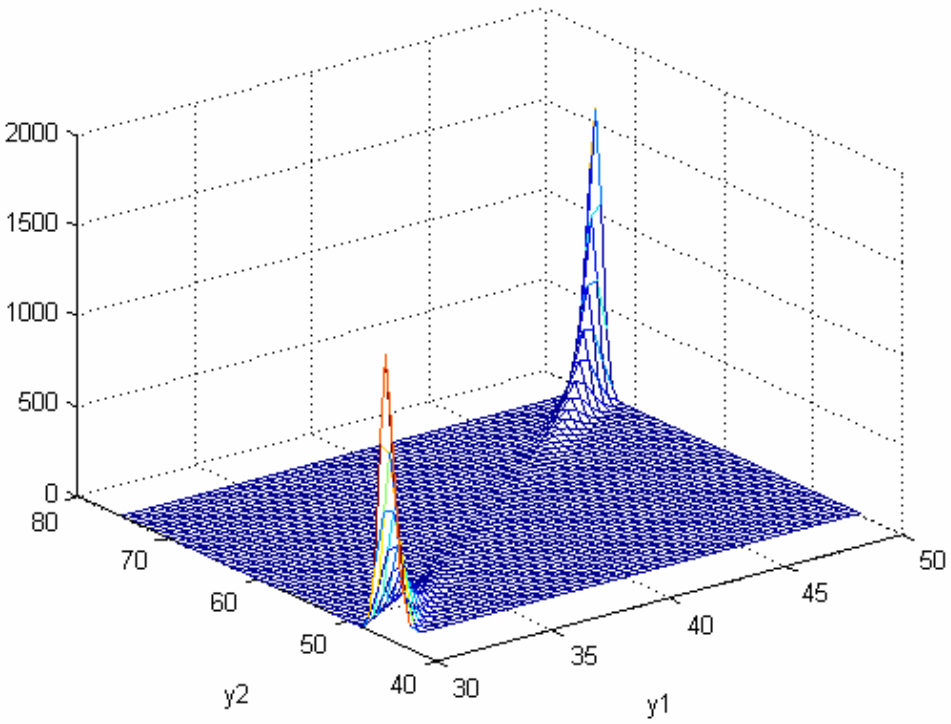


**Figure 27:** NPDF joint randomness term (g-function).

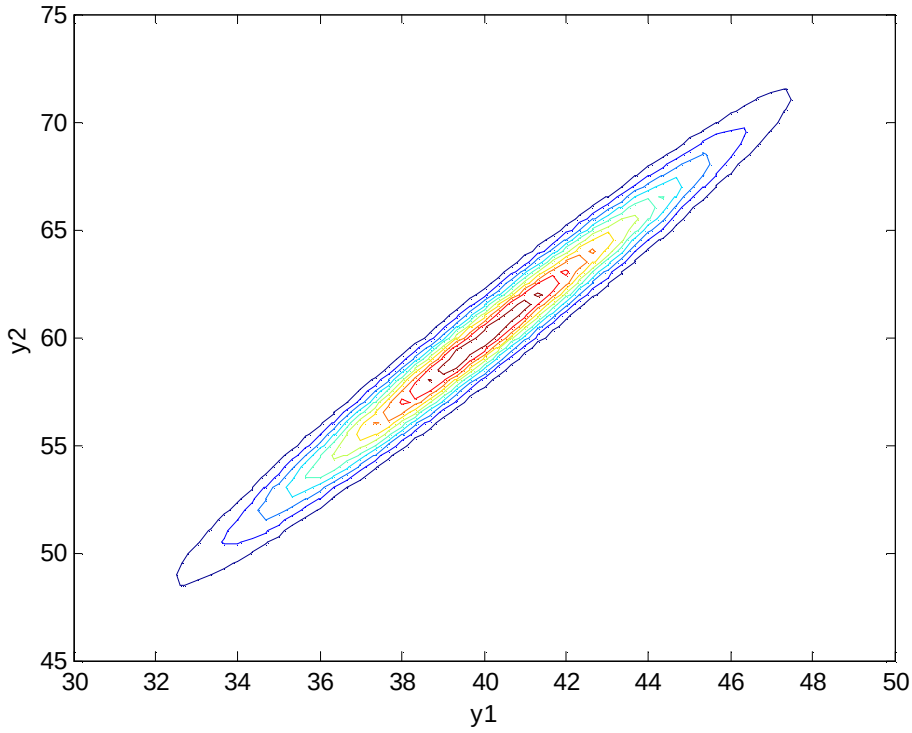


**Figure 28:** Linear System Joint Distribution Using IPDF (Joint Normal Input).

The joint distribution of the input variables is given by the bivariate normal distribution function (equation (3))

$$f_{X_1,X_2}(x_1,x_2) = \frac{1}{2\pi\sigma_{x_1}\sigma_{x_2}\sqrt{1-\rho_{x_1,x_2}^2}}$$
$$exp\left\{\frac{1}{2\rho_{x_1,x_2}^2-2}\left[\left(\frac{x_1-\mu_{x_1}}{\sigma_{x_2}}\right)^2 - 2\rho_{x_1,x_2}\left(\frac{x_1-\mu_{x_1}}{\sigma_{x_1}}\right)\left(\frac{x_2-\mu_{x_2}}{\sigma_{x_2}}\right) + \left(\frac{x_2-\mu_{x_2}}{\sigma_{x_2}}\right)^2\right]\right\}$$

Substituting the Jacobian, $J$, and inverse solution of $x_1$ and $x_2$ into equation (53) yields

$$f_{Y_1,Y_2}(y_1,y_2) = f_{X_1,X_2}(2y_1-y_2, -y_1+y_2) \tag{66}$$

The joint distribution solution using the IPDF method is given by Figure 28 and is also an exact replica of the solution generated using the EDF method.

By inspection of equation (66) one can see that the analytical solution is parametric with respect to the properties of the input distributions, and through a symbolic

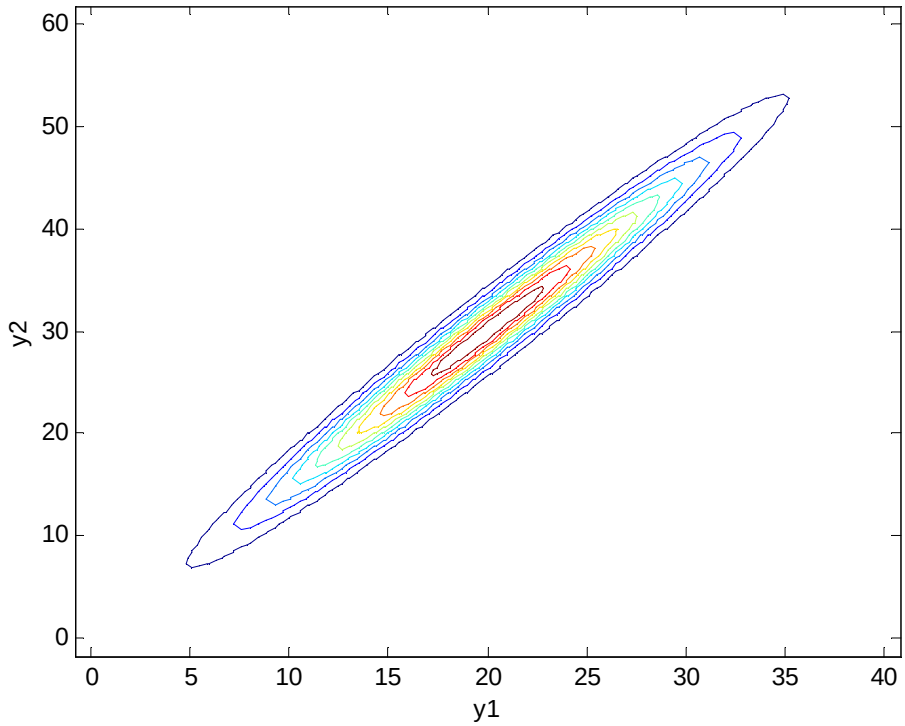


**Figure 29:** Linear System Joint Distribution Using IPDF (Joint Normal Input).

representation of the Jacobian can also be expressed with respect to the linear system coefficients. For instance, assume temporarily that $\sigma = 4$ and $\mu = 10$. The new JPDF, shown in Figure 29, then would require only a simple substitution into equation (66) without any further mathematical operations. For this variant of the problem, the only visual effect was a shift of the mean value. Thus, a truly parametric JPDF of the two responses is created where for the distribution is explicitly given for any real values of the mean, variance, and correlation coefficient of the two input variables.

In this particular problem, the IPDF method provides the exact analytical solution for the joint probability model of the two responses. Thus, the IPDF model can be used to validate the other methods as well as provide their necessary input, circumventing the need to run lengthy simulations. For instance, the marginal distribution

for both responses can also be determined using equation (66) given as

$$f_{Y_1}(y_1) = \int_{-\infty}^{+\infty} f_{Y_1,Y_2}(y_1, y_2)dy_2$$

$$f_{Y_2}(y_2) = \int_{-\infty}^{+\infty} f_{Y_1,Y_2}(y_1, y_2)dy_1$$

Also, by using equation (6) the correlation coefficient can be computed and is found to be 0.98 which is in perfect agreement with the baseline EDF solution.

Additionally, the joint randomness term in either the BPDF or NPDF model can be validated using the IPDF method. By starting with an assumed general JPDF model set equal to the JPDF solution provided by IPDF, one can solve for the exact g-function as follows

$$g(y_1, y_2) = \frac{f_{X_1,X_2}(2y_1 - y_2, -y_1 + y_2)}{f_{y_1} \cdot f_{y_2}}$$

where the denominator is the product of the marginal distributions of the responses, $y_1$ and $y_2$, found earlier. This function is shown graphically in Figure 30. The actual g-function is apparently a saddle function and is accurately predicted, as shown in Figure 25, using the NPDF method. However, for this case the BPDF g-function does not predict the actual g-function behavior very well as the magnitude of the off center weighting shown by the two peaks is too small.

### 3.6.2 Linear Responses with Independent Non-normal Input

The second case considered for evaluating the various JPDF models is that of the same linear system described previously but with independent lognormal input variables with $\mu = 20$, $\sigma = 2$ ($\mu_{LN} = 3$, $\sigma_{LN} = 0.1013$). Failure is defined to occur when either $y_1 \geqslant 46.5$ or $y_2 \geqslant 70$. The actual input and response joint distributions are given by Figure 31 and Figure 32, respectively. The response distribution was found using the EDF method which was also used to compute the failure probability of this problem to be 0.02364.

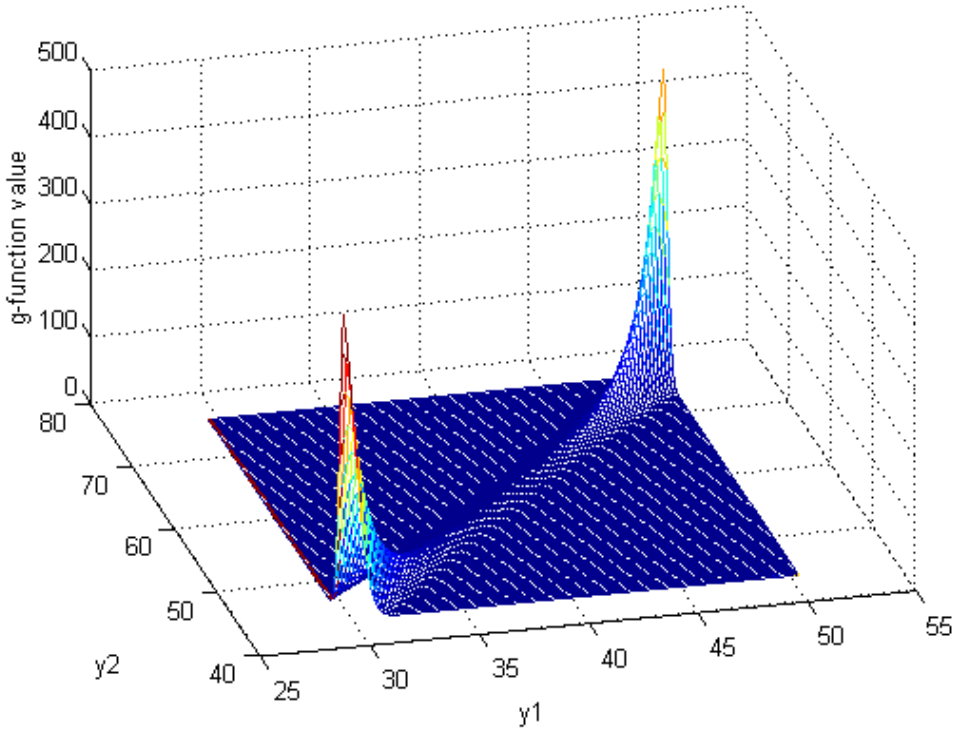


**Figure 30:** Actual Linear System *g-function*.

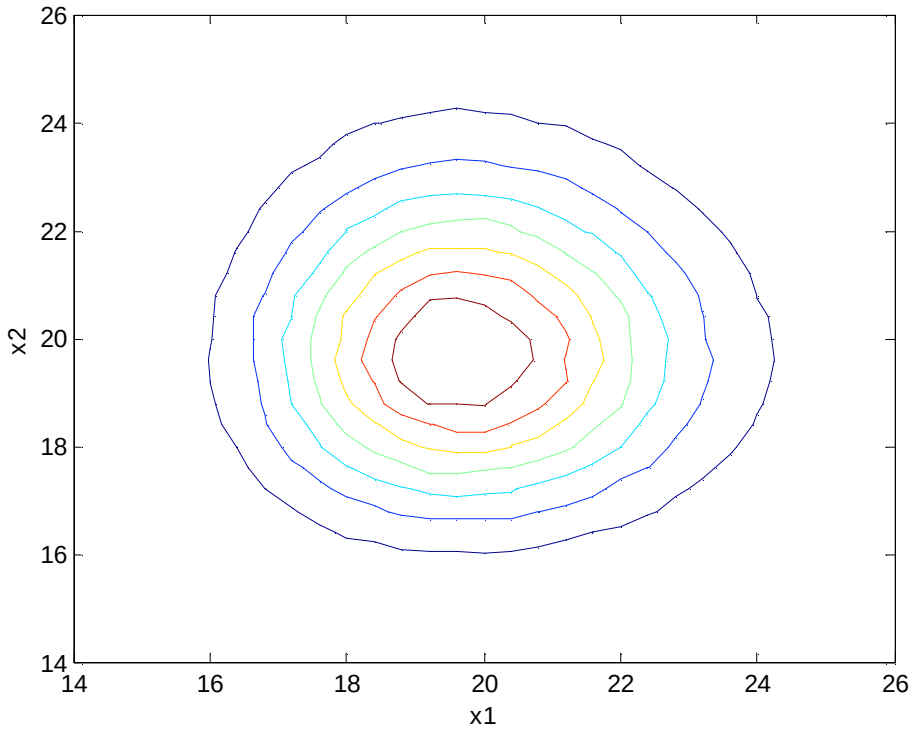


**Figure 31:** Lognormal Input Distribution Space.

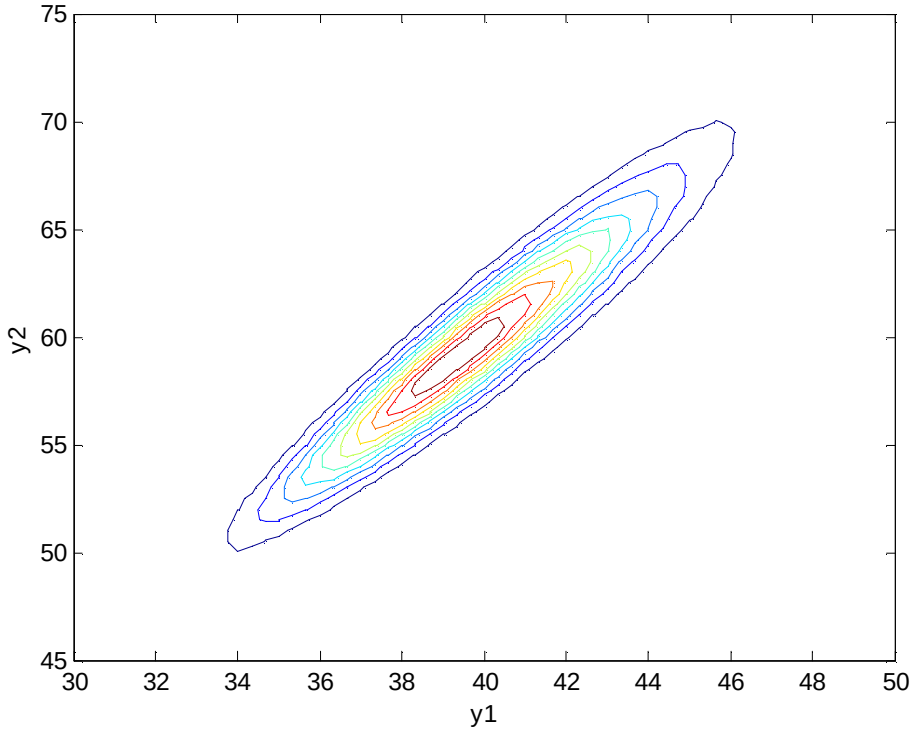


**Figure 32:** Linear System Joint Response Distribution (Non-normal Input).

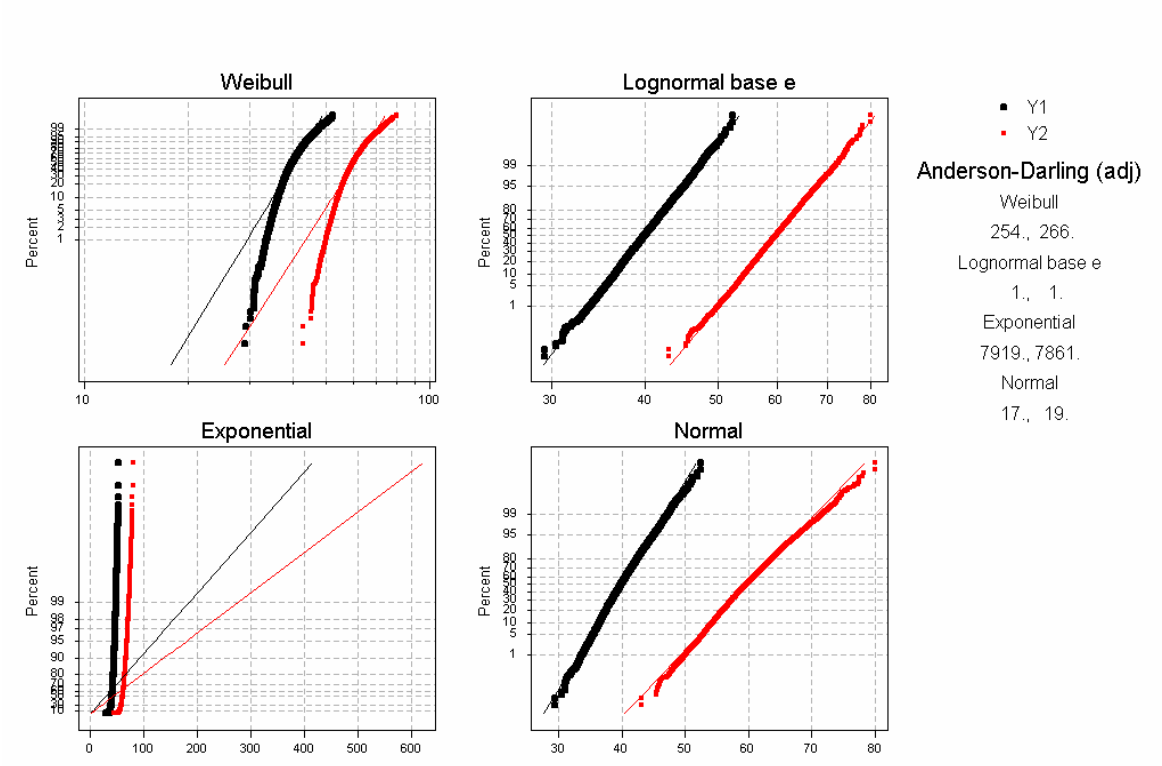


**Figure 33:** Identification of Linear Response Marginal Distributions (Non-normal Input).

This case is interesting because, although the lognormal input is independent, the resulting response distributions are not only lognormal (see Figure 33) but exhibit strong mutually dependence as the correlation coefficient was found using the EDF method to be 0.95. Figure 33 is a probability plot showing how well each of four distributions fit the response data. The lognormal distribution most accurately represents the data for this case.

The solution of the probability of failure as calculated using each of the five methods explored in this study is reported in Table 3. The results reported again demonstrate the importance of considering the joint randomness of functions that share the same random variables as input. Again, both the IPDF and NPDF accurately predict the actual failure probability solution. As expected, the MFOSM method produces

**Table 3:** Linear System Failure Probability Results (Non-normal Input).

| | Method | | | | | |
|---|---|---|---|---|---|---|
| **Output** | **EDF** | **FPI** | **BPDF*** | **MFOSM** | **NPDF*** | **IPDF** |
| Independent | 0.03531 | 0.03027 | N/A | N/A | N/A | N/A |
| Joint | 0.02364 | N/A | 0.02330 | 0.01562 | 0.02332 | 0.02374 |
| Evaluations | 1E6 | 3 | 1E6 | 5 | 1E6 | 5 |

*Using EDF Method for input

a less than accurate solution. This is a result of the joint normal distribution assumption inherent with the MFOSM method. The MFOSM solution for this case is the same as the independent input solution of the previous case for a linear system with joint normal input. Thus, the advantage of the IPDF method in accounting for non-normal input is demonstrated.

### 3.6.3 Quadratic Responses with Joint Normal Input

The last scenario to be utilized for exploring and validating the joint probability model approaches is that of a quadratic system of equations. The objective is to compute the reliability of a component limited by two failure mechanisms represented by the following two responses given as

$$\begin{aligned} y_1 &= x_1^2 + x_2^2 \\ y_2 &= x_1^2 + 2x_2^2 \end{aligned}$$

where $x_1$ and $x_2$ are dependent normal random variables with $\rho_{X_1,X_2} = 0.5$, $\mu = 20$, and $\sigma = 2$. Failure occurs when either $y_1 = 1000$ or $y_2 = 1500$.

The relationship between both of the responses and the input variables is shown in Figures 34 and 35. The solution of the probability of failure as calculated using each of the five methods explored in this study is reported in Table 4. As with the previous two validation cases, the EDF method was used to compute the baseline probability of failure to be 0.10280. The joint distribution functions solved using

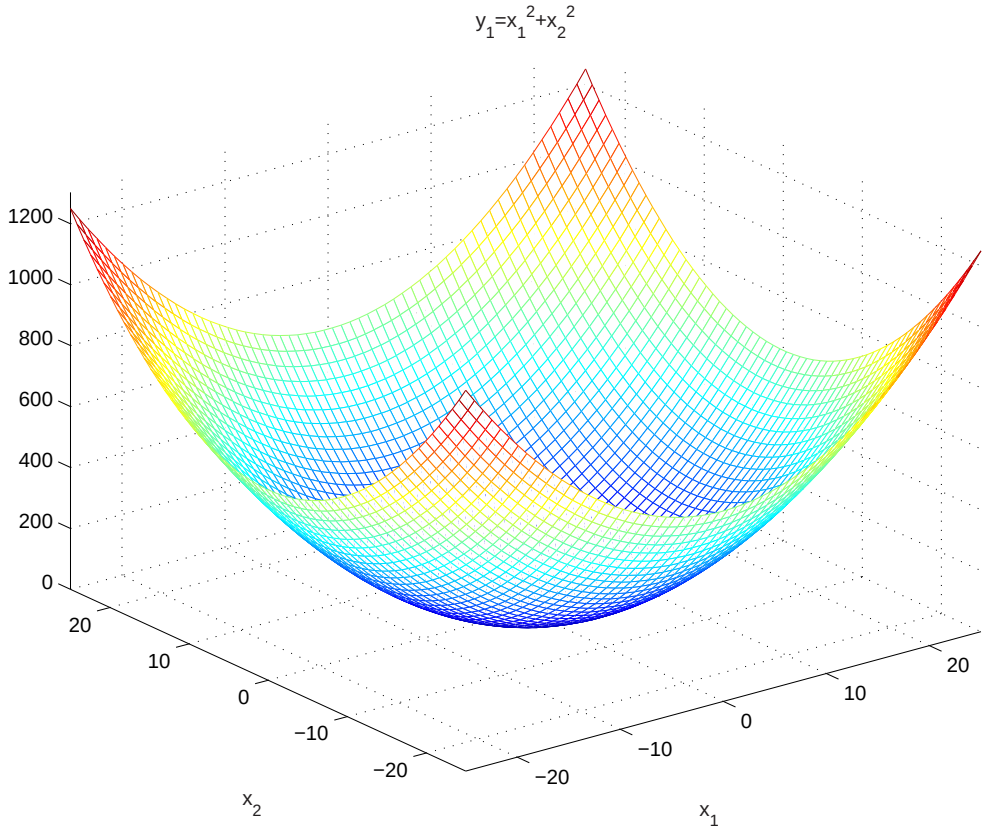


**Figure 34:** Quadratic System $y_1$ response.

**Table 4:** Quadratic System Failure Probability Results.

| | | Method | | | | | |
|---|---|---|---|---|---|---|---|
| **Output** | **Input** | **EDF** | **FPI** | **MFOSM** | **IPDF** | **NPDF*** | **NPDF**** |
| Ind. | Ind. | 0.10839 | 0.10455 | N/A | N/A | N/A | N/A |
| | Corr. | 0.17227 | 0.16816 | N/A | N/A | N/A | N/A |
| Joint | Ind. | 0.07103 | N/A | – | – | – | – |
| | Corr. | 0.10280 | N/A | 0.09400 | 0.10256 | 0.09018 | 0.11993 |
| Evaluations | | 1E6 | 3 | 5 | 5 | 5 | 1E6 |

*Using IPDF for marginal distributions and correlation matrix
**Using EDF for marginal distributions and correlation matrix

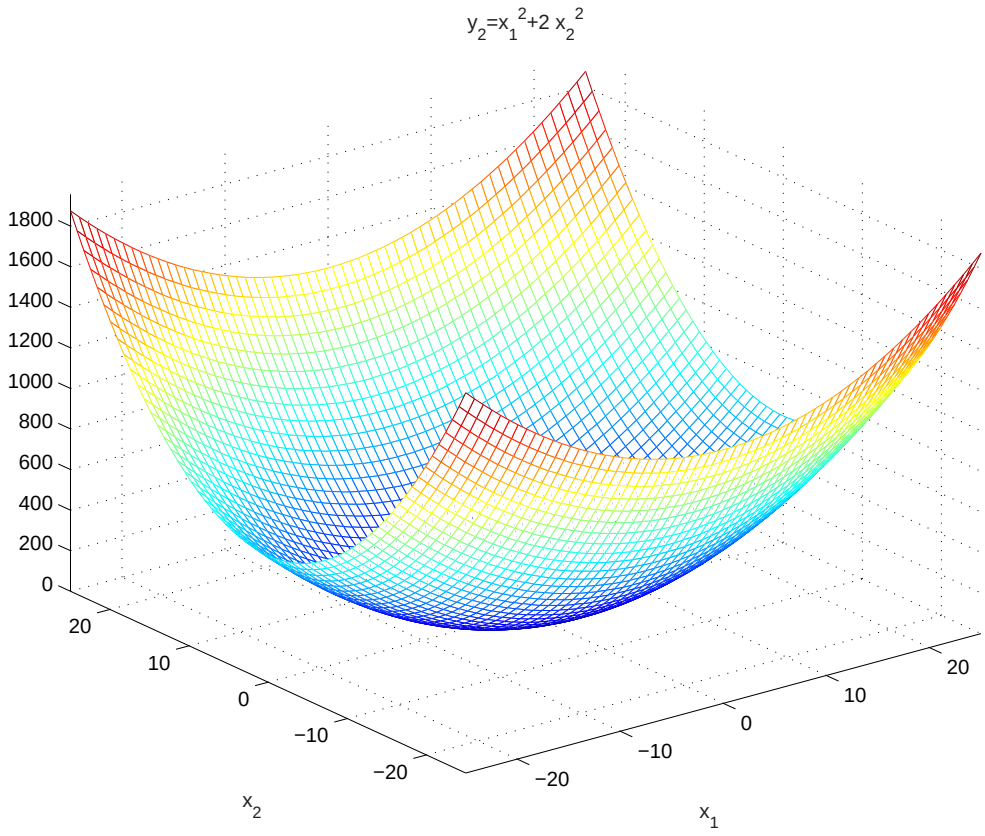


**Figure 35:** Quadratic System $y_2$ response.

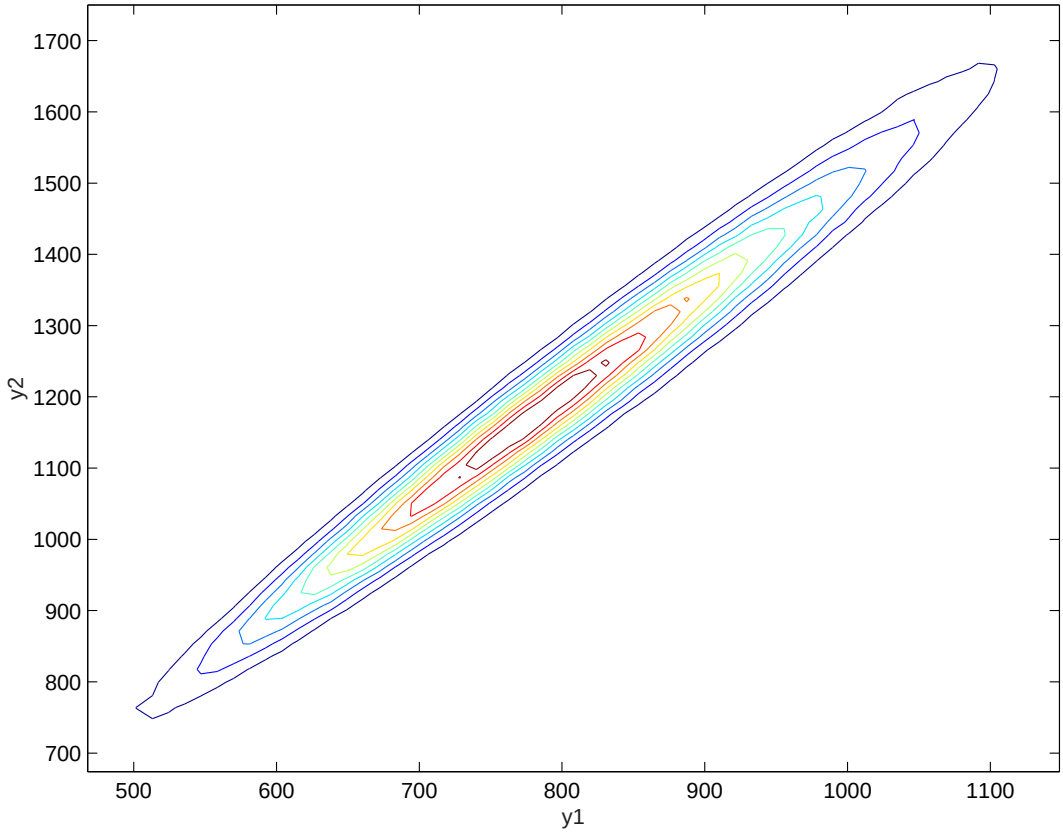


**Figure 36:** Quadratic System Joint Distribution Function Contour Plot.

EDF are given in Figures 36 and 37. An interesting result was obtained. When considering independent input and output, the probability of failure was calculated to be very close to that of the actual jointly distributed case. Thus, the FPI method at first glance would appear to provide a much more accurate result than either of the joint probability methods explored except that the result is just coincidence as it is only a function of the independent solution value and not the FPI method's accuracy involving jointly distributed responses.

Each of the joint probability methods were applied in the same way as the previous examples, except for the MFOSM and IPDF methods. Since the IPDF method requires a linear system of equations with an invertible solution, a Taylor series approximation of the two responses using equation (8) was created and used as input to both the MFOSM and IPDF methods. An interesting result was obtained, with

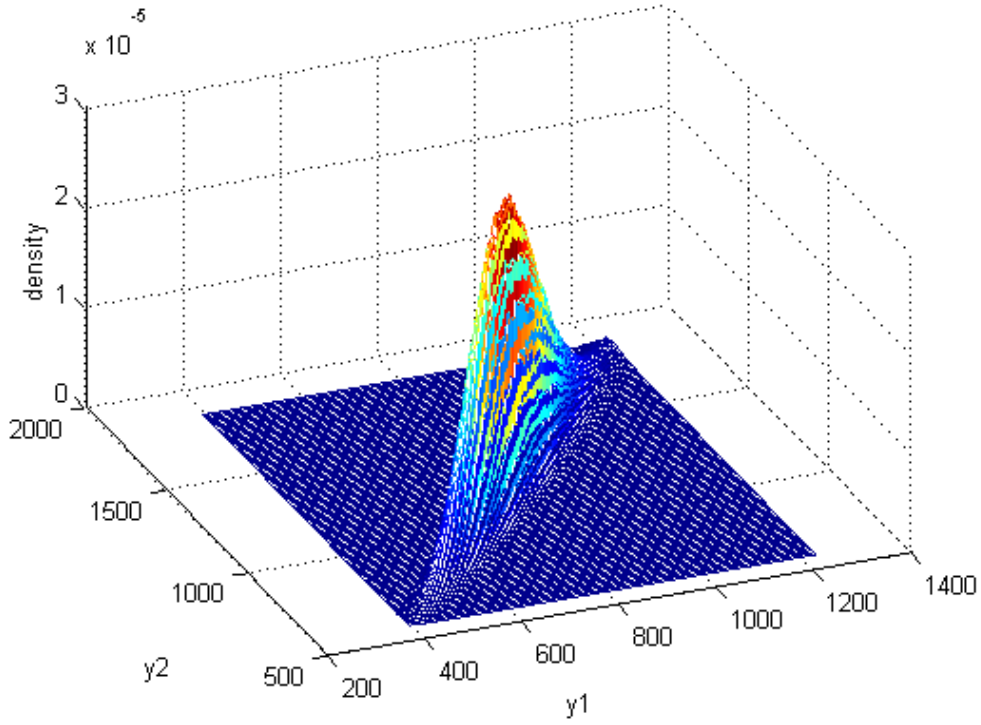


**Figure 37:** Quadratic System Joint Distribution Function Mesh Plot.

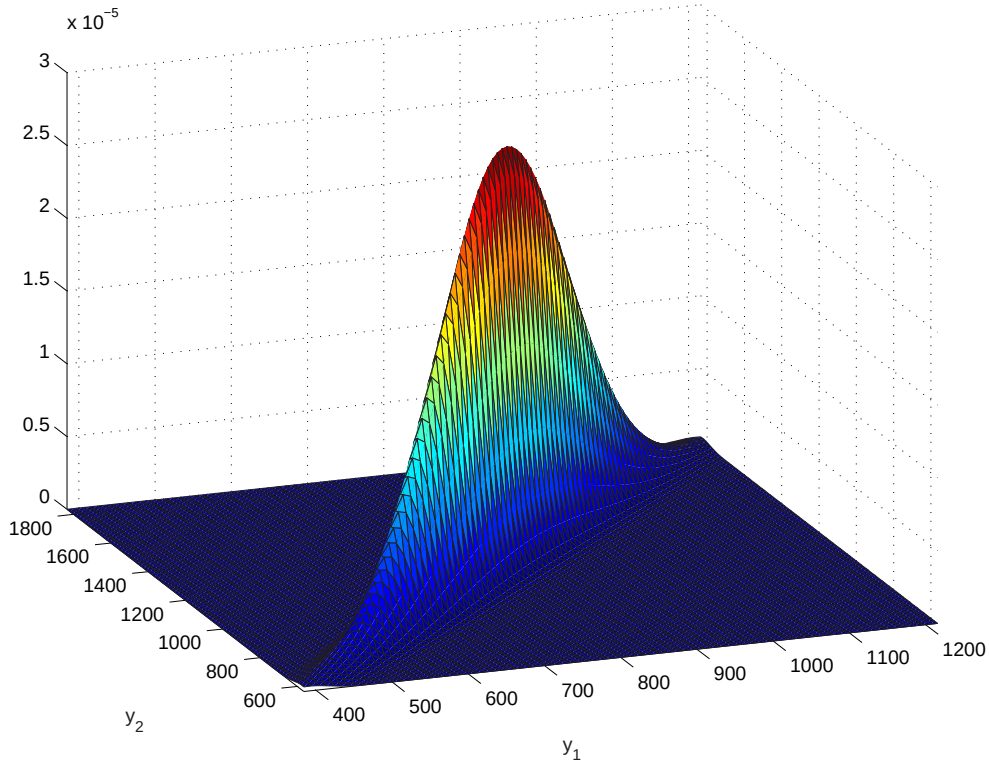


**Figure 38:** Quadratic System Joint Distribution Function Using IPDF.

only five evaluations of the response functions the IPDF method solution was within 0.2% of the actual solution. However, the MFOSM method, which used the same sampling information as the IPDF method, was only within 8.6% of the actual solution. The NPDF method using the sampling information or EDF as input faired even worse than either of the two sensitivity-based methods. The joint probability response predicted by the IPDF method is shown by Figures 38 and 39.

The NPDF method was executed twice for this example; once using the IPDF method as input and again using the EDF method as input. The first execution was conducted by using the marginal distributions produced by the IPDF method and solving for the appropriate $\rho'$ to enable the NPDF joint probability model to be constructed. The computed failure probability was within 12% of the actual solution found using EDF.

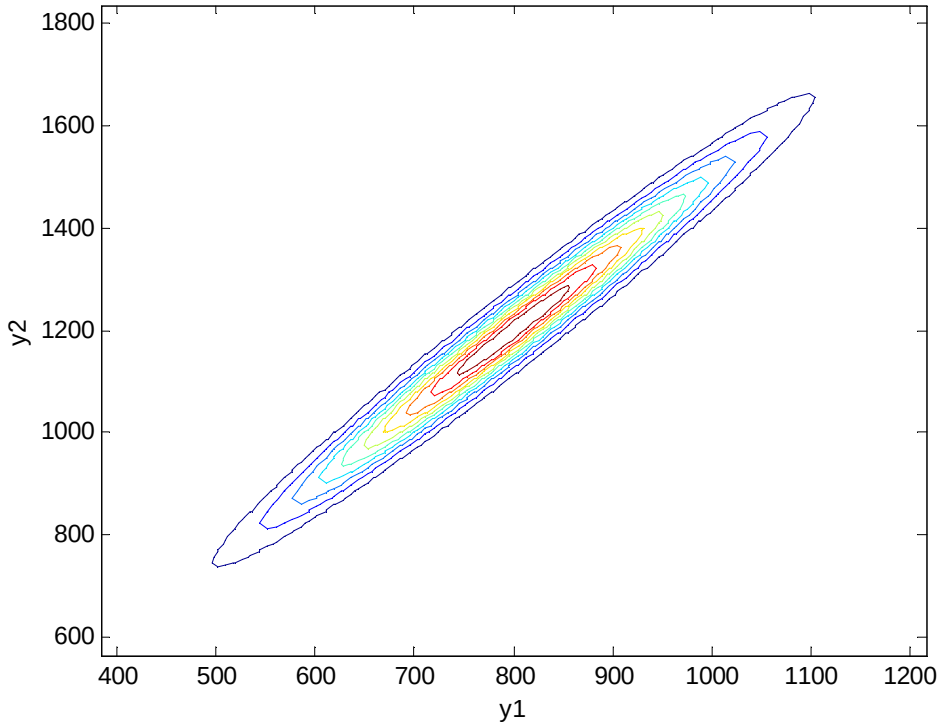


**Figure 39:** Quadratic System Joint Distribution Function Using IPDF.

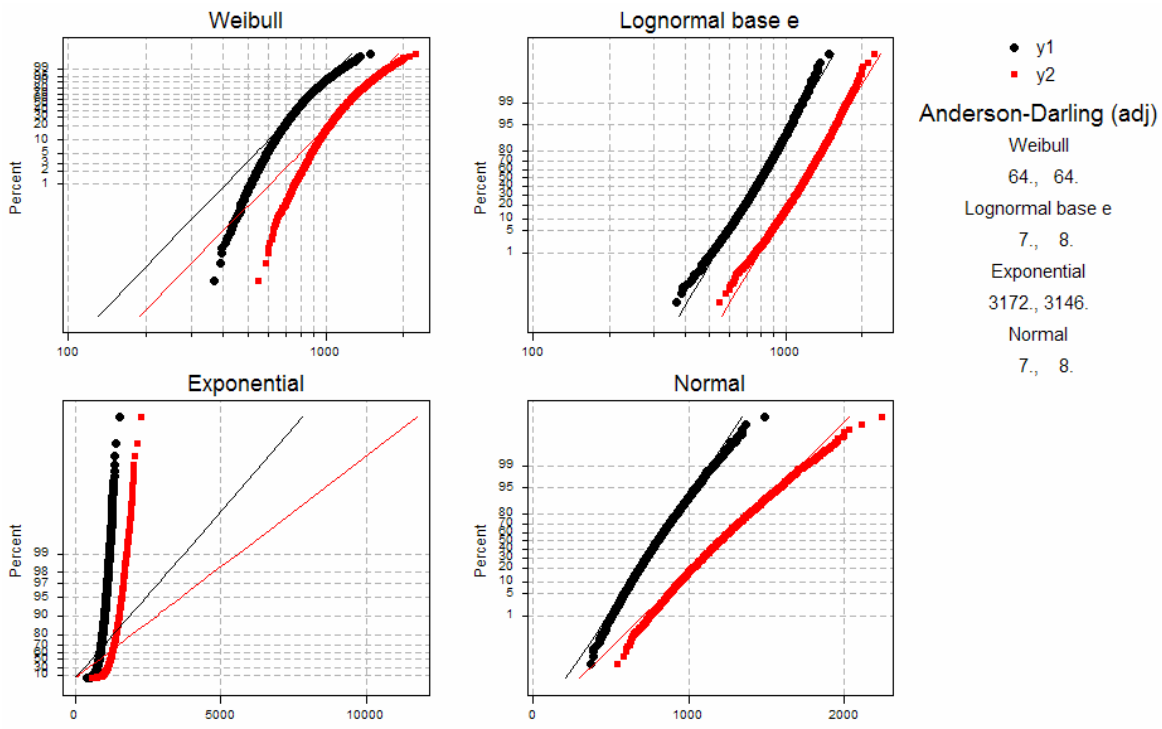


**Figure 40:** Identification of Quadratic Response Marginal Distributions.

The NPDF method using the EDF results is now described. The actual marginal distributions of the two response can be determined using the empirical simulation data by considering several potential parametric distributions and employing the Anderson-Darling test statistic (see section 2.4.1) to determine the most appropriate generic distribution. The probability plots of each data set using each candidate distribution as well as the corresponding Anderson-Darling test statistic are shown in Figure 40. Notice that both the lognormal and normal distribution functions fit the data reasonably well. However, one can conclude based on the comparison between the failure probability solutions and the A-D test statistics for the parametric distributions that the response space appears to follow an atypical distribution that is similar to normal or lognormal. However, for the empirical characterization of the response marginal distributions the normal distribution function is chosen with the mean and standard deviation found to be 808.6 and 139.8, respectively, for $y_1$ and 1213.1 and 213.8, respectively, for $y_2$. Also, the empirically determined correlation

**Table 5:** Quadratic System IPDF Method Results.

| | **Independent $x_i$** | | **Dependent $x_i$** | |
|---|---|---|---|---|
| **Statistics** | **EDF** | **IPDF** | **EDF** | **IPDF** |
| $\mu_{y_1}$ | 808.1 | 800.0 | 807.9 | 800.0 |
| $\mu_{y_2}$ | 1212.1 | 1200.0 | 1212.0 | 1220.0 |
| $\sigma_{y_1}$ | 113.4 | 113.1 | 138.8 | 138.6 |
| $\sigma_{y_2}$ | 179.4 | 178.9 | 212.1 | 211.7 |
| $\rho_{y_1,y_2}$ | 0.95 | 0.95 | 0.98 | 0.98 |
| $P_f$ | 0.07103 | 0.05732 | 0.10280 | 0.10256 |

coefficient between $y_1$ and $y_2$ was found to be 0.98 using the EDF data. The NPDF method was then re-executed with the actual partial distribution information and the probability of failure re-computed as 0.11993 which is within 17% of the actual solution. Thus, the information gleaned using the IPDF method is more representative of the actual response distribution even though a linearization of the deterministic space was used. Apparently, for this problem the joint randomness behavior was more of a driver in the failure probability solution than the deterministic functional relationship.

Finally, system reliability methods [45] can be used in concert with the structural reliability methods, discussed in section 2.3, to compute an approximate joint response failure probability solution. Using the system reliability method and the MPP reliability indices, the probability of failure (correlated input, joint output) was found to be 0.8989 which is as accurate as the IPDF method. The disadvantage of this result is that it is completely unique to the input variable statistical values and is not parametric. But, using the MPP points one can find the joint most probable point (JMPP) and use it as the point in which the Taylor series approximation is conducted for the IPDF method. Combining the JMPP calculation with the IPDF method would be an interesting approach to pursue for future work related to this study.

## 3.7 Summary

Identifying and quantifying the marginals as well as computing the correlation coefficient would be the typical industrial procedure for an actual vector of responses such as that demonstrated using this example. However, numerous evaluations of the responses would be required at a considerable computational expense unless this information was assumed. As shown by the validation problems the sensitivity-based methods, MFOSM and IPDF, produced a more accurate result with considerably less effort than the alternative joint probability models. Further, it remains to be shown but intuition could be used to reason that an even greater benefit could be expected in using the IPDF method for more unusual cases with respect to the statistical behavior of the input variables. Also, the results of the MFOSM and IPDF methods can be utilized successively across various levels of a system represented by functional relationships. At each level the joint probability distribution of the responses could then be utilized as the joint input distribution of the next level under consideration. The BPDF method did not properly capture the joint randomness behavior for the cases considered. This deficiency is attributed to the g-function used with this method as it does not account for asymptotic behavior of the actual g-function. The other sample-based method, NPDF, proved to be a useful constructive JPDF model. However, many samples of the response space are required to specify the necessary NPDF distribution and correlation matrix input. Therefore, the sensitivity-based methods developed and demonstrated within this study are recommended as enabling models for the formulated framework discussed in the following chapter.

# CHAPTER IV

# FRAMEWORK FORMULATION

The statistical behavior of a component can be altered significantly when it operates within a complex system. This is analogous to differing behavior of a structural member in isolation versus during operation in its parent system. Conventional probabilistic approaches neglect this phenomenon resulting in, the best case, limited and, the worst case, grossly inaccurate results. A likely source of this dilemma is the rather complex and inadequately considered statistical environment within which the component operates. Now that the relevant background theory involving stochastic, physics-based reliability assessments has been provided, and a thorough evaluation of available joint probability modeling has been conducted, an improved framework that addresses current limitations of component reliability assessments is given. Implementation of the framework is discussed in Chapter 5.

## *4.1 Criteria*

The requirements, or criteria, of the framework are as follows. The framework must seamlessly integrate stochastic theory and physics-based models for improved probabilistic analysis of components of complex systems. It should provide for early characterization of the internal stochastic behavior of intermediate quantities given stochastic global and local input. Such a characterization inherently should account for the joint probability of the necessary intermediate variable and response quantities in addition to any significant non-Gaussian[1] behavior within the system of interest.

[1]Gaussian is synonymous with Normal in terms of continuous distributions

Efficiency of the framework is required for practical implementation. The framework should be amenable to the system design process and therefore consistent with conventional multi-disciplinary design and optimization techniques. Several features necessary to meet these requirements are summarized as follows:

- Amenable to current industrial life prediction practices
- Capable of accounting for complex joint randomness behavior of component reliability input
- Interface to modern reliability methods
- Efficient use of existing physics-based engineering models
- Parameterized approach with respect to design and control variables

## 4.2 Overview

A solution to these requirements in the form of a six step framework is described within this chapter. A portion of the framework is an ensemble of highly useful, but sometimes overlooked methods found in several disciplines, including: systems engineering, system safety, structural reliability, statistical reliability, multivariate statistics, and joint probability. A thorough review of the available literature within each of these areas provided only a portion of the elements and corresponding tools required to create the framework. New tools for modeling joint randomness, described in chapter 3, had to be developed, such as new joint probability distribution models and efficient joint statistical estimation using the covariance approximation technique (Appendix A).

The framework consists of following six major steps: identification, decomposition, synthesis, characterization/quantification of the local statistical space, parameter space reduction, and probabilistic assessment. These steps are graphically depicted in Figure 41. The initial step of *identifying* the component(s) to analyze is one

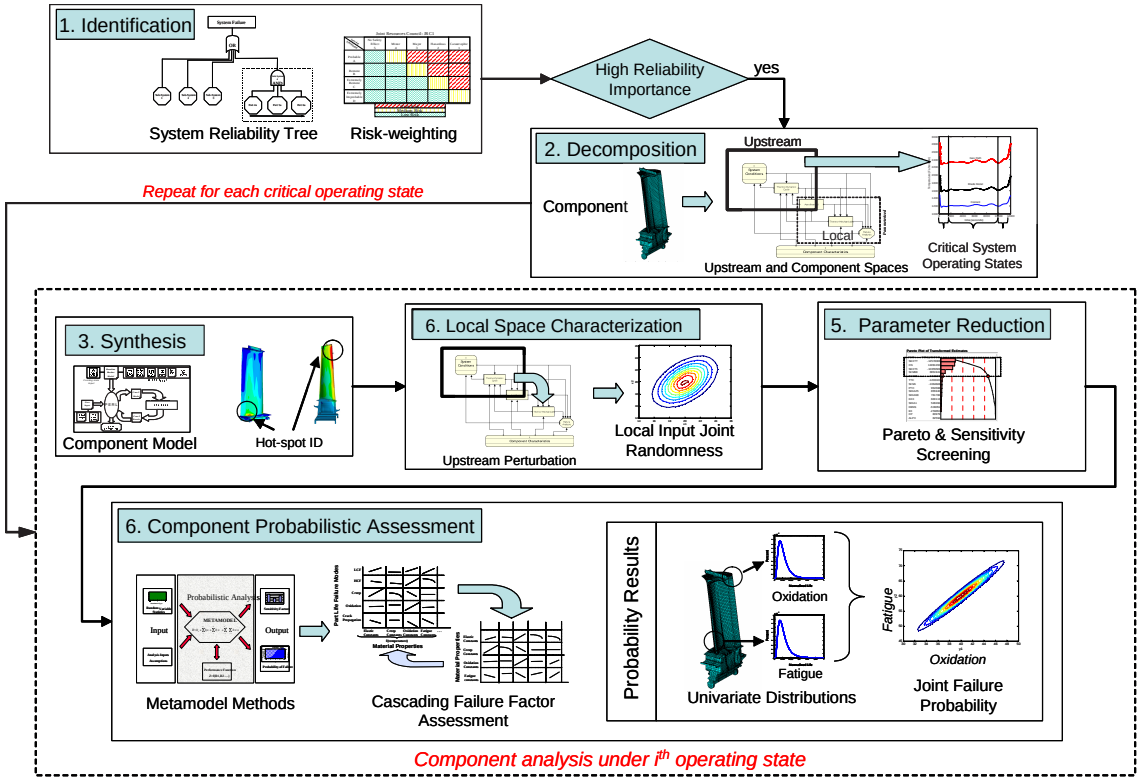


**Figure 41:** Framework Steps.

that is more or less implied in any component reliability study, however, it is worthwhile to include for completeness of any such framework. An improved component selection approach is prescribed as an alternative to the traditional 'expert opinion' approach. Often the primary components are selected based on engineering judgement which in turn is usually based on experience and intuition. However, for a new system the relative importance of a particular component might change significantly. In addition, with changing customer needs, other factors such as cost might require consideration.

The next step is to take the available information on the component and its intended environment and *decompose* this information to a more useful form. The objective of this step is to reveal the structural and statistical importance of the component as well as gain a preliminary understanding of its behavior. This behavior would include the applicable failure modes and all necessary contributing analyses and

input. Next, using the information derived during the decomposition, an analysis scheme is constructed during the analysis *synthesis* step. The analysis environment would include a properly defined and linked set of contributing failure analyses in addition to all required input.

A vital step in the framework is the stochastic *characterization* of the input local to the component reliability analysis. The reasoning behind this local characterization is as follows. The upstream analyses, depicted in the decomposition step in Figure 41, in practice would be extremely time consuming to evaluate for the purpose of quantifying the local input statistical properties. Yet, these properties are extremely important to the accurate assessment of the reliability of the component. The methods explored in Chapter 3 are recommended to efficiently characterize this local input space.

Component analyses typically are highly intensive with respect to the complexity of the analyses in addition to computational demands. Thus, reducing the dimensions to be considered, especially dimensions of statistical variation, is extremely beneficial and is accomplished during the *parameter reduction* step in the framework. The parameter reduction step is recommended after quantifying the local input statistical space. The reason for this is that the parameter reduction entails running a limited set of evaluations of the component failure analysis to determine which, if any, input variables have a negligible affect on the failure analysis. However, a deterministic versus a probabilistic sensitivity analysis could produce a very different ranking of the most important input variables. Thus, having statistical information going into the parameter reduction will allow the engineer to consider each input variable's statistical properties.

With a refined component parameter and failure analysis space, a sound, efficient, and accurate probabilistic failure assessment can be conducted, which is the last step of the framework. Again, the methods of Chapter 3 can be employed to transform the input joint probability space to a form that can be used as input to the modern

structural reliability methods discussed in section 2.3. The joint probability methods explored and developed in Chapter 3 can be used to create a parametric model of the local input joint probability space as a function of top-level system parameters. Therefore, the parametric input space and the subsequent component probabilistic analysis can be evaluated efficiently across numerous conditions. Each of these steps are now discussed in detail.

## *4.3 Identification*

*On which components should a probabilistic assessment be conducted?*

For a complete framework, a means of selecting a component during the design or analysis phase for further physics-based reliability assessments is advantageous. A system-level perspective can greatly improve the ability to select components in a justifiable way. Traditionally, there are several general approaches used to select components for reliability assessments. They include cost-based, failure severity based, likelihood based, or approaches based on field experience. However, rather than compare these approaches and recommend which of them to use with the formulated framework; a more useful approach is given that is flexible enough to incorporate important aspects of many of these approaches. This combined approach is highly useful when considering a new system or an existing system where its usage criteria might change. These situations can quickly push the system design outside of the realm of current engineering experience and judgement. Several importance measures are now given along with a few modifications necessary to provide the elements of an overall component importance measure that involves multiple weighted importance measures.

Borrowing from system safety and traditional reliability methods, a useful starting point for the component selection scheme is a system failure event structure function such as the system fault tree shown in Figure 42. The initial system failure event

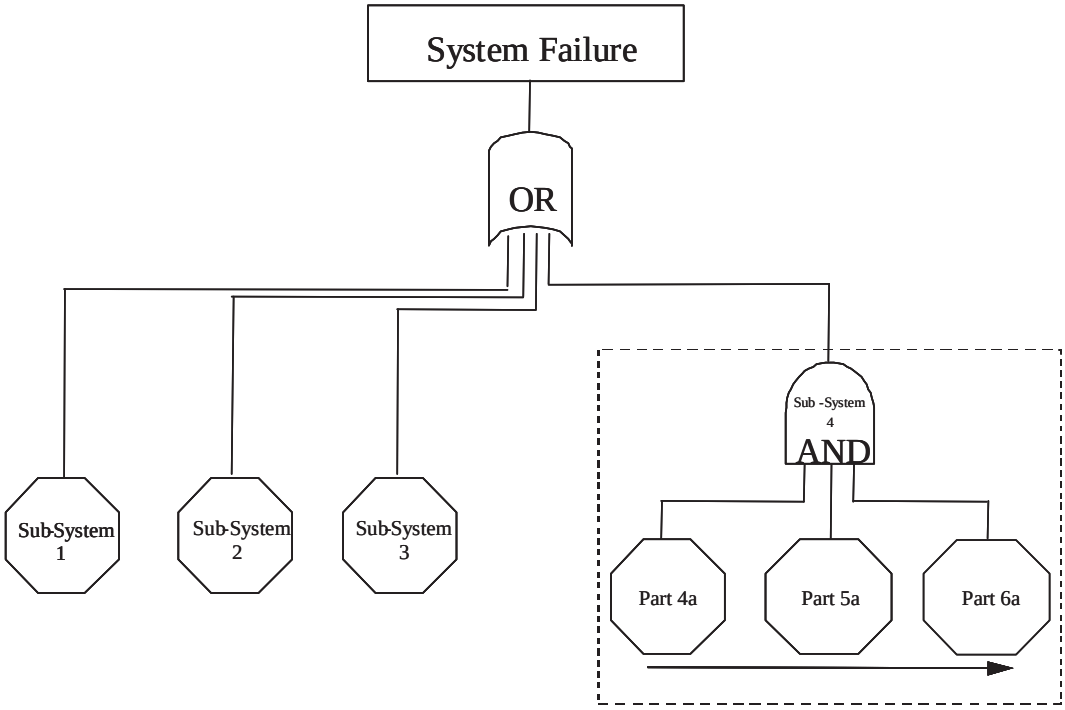


**Figure 42:** System Fault Tree

functions to consider would be those that could lead to a catastrophic failure of the entire system. The FHA tool mentioned in section 1.2 is a useful one for qualitatively ranking these functions to produce a complete likelihood versus severity categorization such as that shown in Figure 43. Each failure event is assigned a numeric value from a set scale according to the likelihood, or probability, that it will occur and, using the FHA results, is assigned a value representing the severity of the event occurring. Once the appropriate severity and likelihood values are assigned to each event, they then can be organized into several levels of risk. Where risk is often defined as the chance of an event occurring multiplied by the severity of the event. Three levels of risk are shown in Figure 43; high, medium, and low. For instance, an event that had a moderate to high likelihood and a major to catastrophic failure consequence, or severity, would be classified as moderate to high risk.

A structural representation of the failure structure function shown in Figure 42 could be comprised of several levels of series and parallel events tying the top-level

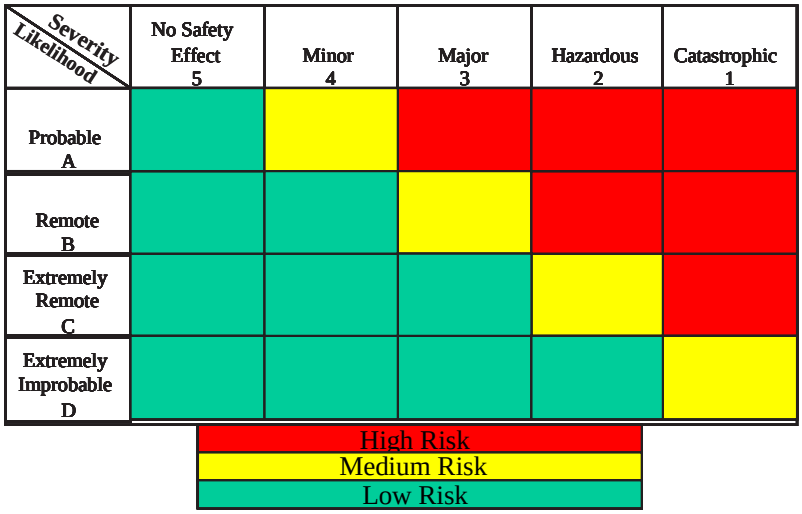


**Figure 43:** Risk Weighting Scheme

failure event with the bottom-level events. Ideally, one would continue adding more levels of consideration below the top-level event until a level is reached in which the bottom events correspond to the failure of a single component. A mathematical failure mapping representing these structural failure conditions is now given.

Beginning with the original formulation of structural importance [53], the binary representation of the functioning condition of the $i^{th}$ component, $x_i$ is

$$x_i = \begin{cases} 0 & \text{if component } i \text{ has failed} \\ 1 & \text{if component } i \text{ is operational} \end{cases} \tag{67}$$

The system state vector, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, yields $2^n$ possible combinations of the system state based on the two possible binary conditions for each of the $n$ components. The structure function, $\phi(\boldsymbol{x})$, is then defined as the mapping between the system state vector and whether the system or top-level event has occurred, given by 0, or has not occurred, given by 1, for the set of system state vectors.

A very useful failure structure function for a $k$ out of $n$ system [48] is given as

$$\phi(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \sum_{i=1}^{n} x_i < k \\ 1 & \text{if } \sum_{i=1}^{n} x_i \geqslant k. \end{cases} \tag{68}$$

which is equivalent to the series system structure function, $\phi(\boldsymbol{x}) = \prod_{i=1}^{n} x_i$, when k=1 and the parallel system structure function, $\phi(\boldsymbol{x}) = 1 - \prod_{i=1}^{n}(1 - x_i)$, when $k = n$. Thus, a complete top-level structure function can be created using $k$ out of $n$, series, and parallel structure functions as building blocks. Finally, a requirement of a structure function at any level is that it is nondecreasing in $x$ and does not improve when a component fails. A structure function that meets this requirement is called *coherent*.

Up to this point, the top-level event structure function has no probability values assigned to it. This is an appealing characteristic for early system level considerations where such values might not yet be available. Yet, the most important components with respect to the top-level event or system reliability can still be found using the structural importance formula. The structural importance for component $i$ and event $j$ [53] is given as

$$I_s(i,j) = \frac{1}{2^{n-1}} \sum_{\boldsymbol{x}|x_i=1} [\phi_j(1_i, \boldsymbol{x}) - \phi_j(0_i, \boldsymbol{x})] \tag{69}$$

for $i = 1, 2, \ldots, n$ in which the summation is over all $2^{n-1}$ state vectors with $x_i = 1$. The summation counts the number of instances of these state vectors which cause the j$^{th}$ top-level event to occur. It is easily shown then that for a coherent system $0 < I_s(i,j) \leqslant 1$. Thus, using this formulation an importance factor related to structural reliability can be computed for any component and top-level system failure event of interest.

### 4.3.1 Weighted Structural Importance

An extension of this semi-quantitative importance function is now given by including weighting and normalization scalars. A weighted composite structural importance with the same range for the i$^{th}$ component can be computed using

$$I_s(i) = \frac{\sum_{j=1}^{m} \alpha_j I_s(i,j)}{m} \tag{70}$$

where $m$ is the number of top-level failure events under consideration and $\alpha_j$ is a weighting factor proportional to the severity of the j$^{th}$ failure event. To maintain the zero to unity range, $\alpha_j$ consequently must also be confined to a range between zero and unity inclusive. Practically speaking, the event weighting factor, $\alpha_j$ would be proportional to the severity of the j$^{th}$ event. This provides a link to the relative functional assessment at the system level produced using the FHA tool. For instance, a catastrophic system failure event could be assigned a weighting factor of unity while a minor event could be assigned a value closer to zero.

The composite structural importance factor given by equation (70) is a deterministic quantity. Modifying this expression to include the likelihood of the component failure is relatively straightforward by again using traditional reliability theory. First, in addition to the coherence assumption for the failure structure, the components are assumed to be non-repairable and their failure likelihoods independent. The independence assumption is a necessary step at this level but will be relaxed later in the overall framework formulation when considering component *input* parameters. Next, the state of the i$^{th}$ component is now represented by a discrete random variable, $X_i$, which is equal to unity if the component is functioning and zero when it is in a failed state. The probability that component $i$ is functioning at a specified time is given by $p_i = P[X_i = 1]$. The probabilities can be represented by the reliability vector $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$. Finally, the top-level system probability for the j$^{th}$ event is defined as

$$r_j(\boldsymbol{p}) = P[\phi_j(\boldsymbol{X}) = 1] \tag{71}$$

where $\boldsymbol{X}$ is the random system state vector for the $n$ components. So, for the case of a series system of components the system reliability under the j$^{th}$ event is simply

$$r_j(\boldsymbol{p}) = \prod_{i=1}^{n} p_i$$

A reliability importance, analogous to the structural importance function given

earlier, can be used to compute the relative importance of each component as a function of its likelihood as well as its structural importance. Using Birnbaum's measure [14] with a modification to denote different top-level events, the reliability importance of component $i$ under the $j^{th}$ top-level event is

$$I_r(i,j) = \frac{\partial r_j(\boldsymbol{p})}{\partial p_i} \tag{72}$$

for $i = 1, 2, \ldots, n$.

### 4.3.2 Weighted Reliability Importance

Further modifying Birnbaum's measure, the composite reliability importance of component $i$ considering all of the top-level events of interest is then given by

$$I_r(i) = \sum_{j=1}^{m} \frac{\alpha_j I_r(i,j)}{m}. \tag{73}$$

This function encompasses not only the functions of interest but their relative importance with regards to the safety and reliability of the system. However, other importance measures might be of interest.

### 4.3.3 Weighted Economic Importance

The economy of failure can be included through a normalized term which is the ratio of the components production cost with that of the system. A cost importance factor might take the following form

$$I_c(i) = \frac{n_i \cdot C_{component}}{C_{system}} \tag{74}$$

where $n_i$ is the number of $i$ components, $C_{component}$ is the item cost of production of the component, and $C_{system}$ is the cost of the entire system.

### 4.3.4 Overall Component Importance Measure

A straightforward way of combining several component importance metrics, such as cost and reliability importance, could be given as follows

$$I^o(i) = \sum_{k=1}^{l} \beta_k I_k(i) \tag{75}$$

where $l$ is the number of importance measures all of a normalized form, $I_k(i)$ is the k$^{th}$ importance measure for the i$^{th}$ component, and $\beta_k$ is a nonzero relative weighting factor for the k$^{th}$ importance measure with a range from zero to unity and the additional requirement that $\sum_{k=1}^{l} \beta_k = 1$. The relative weighting factors provide a control to the significance of each component importance measure. For instance, a value of one half can be assigned to each weighting factor to equally weight each component importance measure. Or, for the two measure case of cost and reliability one might place more emphasize from a safety perspective on the reliability and use a weighting factor of 0.75 and 0.25 for the reliability and cost measures, respectively.

## *4.4 Decomposition*

*What failure modes should be modeled? Which analyses are required? What operating points should be considered?*

Once a component has been chosen for a reliability assessment, a qualitative assessment and decomposition, illustrated in Figure 44, is necessary to determine the appropriate analysis methods and modeling aspects. First, the functional importance of the component must be defined. A useful tool for gaining a better understanding of the functional requirements of an item is the functional relationship diagram shown in Figure 45. As shown in the figure, the component has a function that operates on energy and mass that enters its control volume and thus produces an altered energy and mass state upon completion of the function. The characteristics block represents the properties of the component that are involved with performing the intended

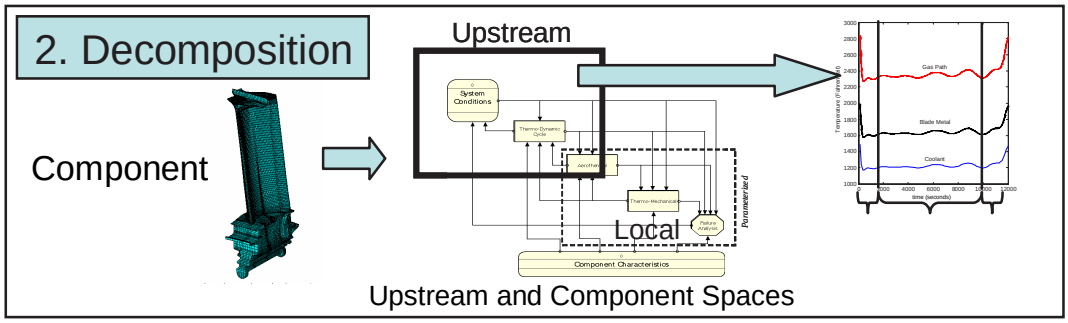


**Figure 44:** Component Analysis Decomposition

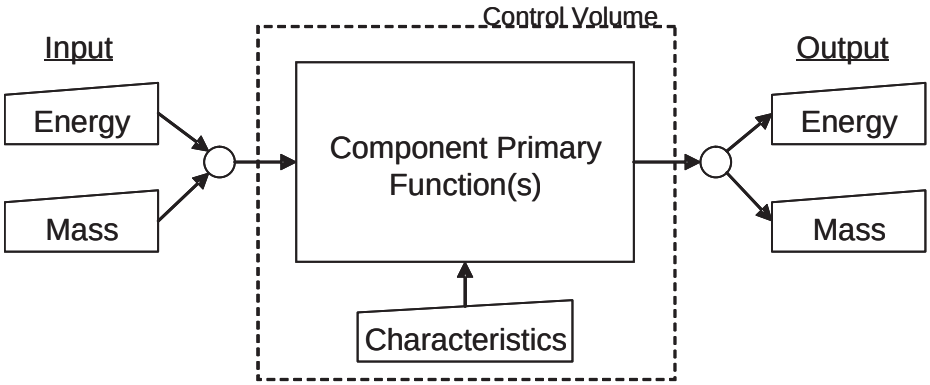


**Figure 45:** Component Functional Relationship Diagram

function. Some typical characteristics, for example, might include geometric and material property variables. In terms of reliability, the possible events that could lead to the inability for the primary functions to be performed should then be identified. A useful arrangement of the functional limiters (i.e. failure modes) related to the functional relationship is given in Table 6. The potential failure modes are listed as rows and crossed with the classes of functions and inputs that can affect the failure mode prediction provided in the columns. The same functional inputs are listed as failure mode inputs since the properties of either class of input can affect both the function and failure properties of the component.

Next, the relevant failure modes are identified. Failure modes in this context would be any phenomena that could lead to the inability of the component to carry out its intended function(s). Each possible failure mode can then be linked to the particular

**Table 6:** Failure Mode Identification.

| | **Function** | | | **Parameters** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | **Mass** | | | **Energy** | | | **Characteristics** | | |
| | F1 | F2 | ... | M1 | M2 | ... | E1 | E2 | ... | C1 | C2 | ... |
| **Mode 1** | | | ... | | | ... | | | ... | | | ... |
| **Mode 2** | | | ... | | | ... | | | ... | | | ... |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

function or functions it can affect as well as the necessary input parameters involved with predicting the failure mode occurrence. The columns of component functions are a direct link to system failure events and therefore the top-level system safety structure. Whereas the traditional identification of relevant failure modes has been accomplished initially using both field experience and engineering judgement, it is done so here in a more congruent manner with respect to top-level system safety assessments.

Aside from using sources such as field experience and engineering judgement, a useful means of identifying potential failure modes would be to conduct a simplified calculation to approximate the local stress, temperature, and excitation vibrations of the component using low-order formulas and appropriate assumptions. These early calculations can be compared to limiting values of the component, namely material and structural constraints such as material strength, buckling load, and natural frequency modes to see which modes might be activated. Representative component loading driven by system operating conditions can be predicted using a low-order approach. The activated failure modes can be predicted at each primary segment of the system operating profile. This amounts to a new functional decomposition for each operating condition and unique sets of failure modes and input parameters. Similarly, the critical operating conditions can initially be screened using a low-order functional/failure assessment.

Once the various potential failure modes and relevant types of input have been identified, further classification of the input should be pursued as it has implications to the synthesis step discussed next. For instance, if an input such as an energy property is time-variant then a deterministic consequence is to consider a transient physical analysis and a non-deterministic consequence would be to consider modeling this input with a stochastic process. Additionally, dissimilar sources of statistical variation require some sort of statistical mixture model which can require an alteration to the implementation of the probabilistic framework. Specifically, the input representing component characteristics will be of a different class of variation than that of the systems in which the component will operate. The energy and mass input to the component functional volume will have a statistical variation driven by other components or sub-systems. A compound distribution function approach is to apply the mixture distribution model given in section 2.4.3. This model can be used to mix these different sources of variation to resolve the component failure distribution. Finally, a control category is considered to decipher whether the input is considered a controlled design variable or a noise variable that is driven by the variation of uncontrollable input. An example of a design variable would be a geometric property of the component while a noise variable could be one of the energy inputs driven from another sub-system within the system of interest. Table 7 is an example of how these characteristics can be organized to facilitate the execution of the framework.

## *4.5 Synthesis*

*Which analyses should be used? What sort of environment is necessary? How is it linked to upstream system analyses?*

With the failure modes defined and all relevant failure mode input described, the next step is to construct a suitable modeling and simulation environment that relates the failure modes to the identified input parameters. A useful information

**Table 7:** Parameter Classification.

| | | Parameters | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Mass | | | Energy | | | Characteristics | | |
| **Category** | **Descriptor** | **M1** | **M2** | ... | **E1** | **E2** | ... | **C1** | **C2** | ... |
| **Behavior** | Static | | | ... | | | ... | | | ... |
| | Temporal | | | ... | | | ... | | | ... |
| | Spatial | | | ... | | | ... | | | ... |
| **Variation** | Component | | | ... | | | ... | | | ... |
| | System | | | ... | | | ... | | | ... |
| **Control** | Design | | | ... | | | ... | | | ... |
| | Noise | | | ... | | | ... | | | ... |

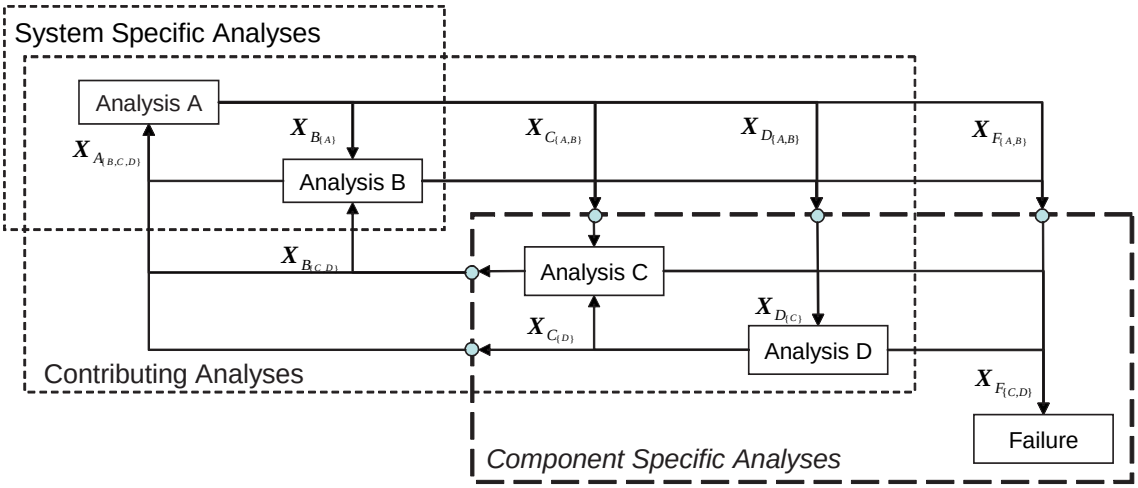


**Figure 46:** Synthesis Schematic

management tool for complex analyses is the analysis structure matrix (ASM). The ASM tool helps the analyst organize the information flow between several analyses that each contribute to the response of the item in question. An ASM of a generic component failure analysis environment is given by Figure 46. Several *contributing analyses*, denoted by the thin dashed perimeter, are used to compute the input for the failure mode analysis. Those analyses that are *component specific* are denoted by the bold dashed perimeter and represent analyses that are conducted at the component level. An example of a component specific analysis would be a mechanical analysis to determine the component stress and strain field. The usual way of describing the flow

of information within this analysis schematic is to start with each analysis, beginning with Analysis A. The A$^{th}$ analysis produces output which becomes the input to one of the other analyses in the order of their arrangement. This is called the feedforward loop between analyses. The feedback loop is created when a subsequent analysis provides input to a preceding analysis represented by the arrows in the lower half of the analysis schematic triangle. A circular or coupled state would be created when there exists both a feedforward and feedback link between a pair of analyses. Any circular condition created by such a relationship would require an iterative procedure to find a solution.

The flow of data, or parameter values, into and out of the component specific analyses module occurs along the perimeter denoted by the circular nodes. The upper boundary of the component specific analysis structure is where the upstream, otherwise described as system-level, input is passed to the component specific analyses. Vectors of input are denoted at each input node. For instance, the input vector $\mathbf{X}_{B_{\{A\}}}$ is the input for the B$^{th}$ analysis provided by analysis A. Therefore, the input vector from the system level analyses to the component specific set would be represented by $\mathbf{X}_{system} = \{\mathbf{X}_{B_{\{A\}}}, \mathbf{X}_{C_{\{A,B\}}}, \mathbf{X}_{D_{\{A,B\}}}, \mathbf{X}_{F_{\{A,B\}}}\}$ where $\mathbf{X}_F$ is the input to the failure mode analyses. This system-component boundary is of particular importance to this study.

The component specific analyses module depicted by the appropriate boundary in Figure 46 deserves an extended discussion. This boundary is one that is commonly used in practice when conducting component life assessments or other component-type analyses. The assumption, statistically speaking, is that the component input statistics are independent of upstream analyses. First, system analyses are executed once to define a baseline determinsitic solution. Variation properties are then prescribed using expert opinion and are provided as input to the component analysis modules. The assumption of statistical independence and subsequent parameter

variances specification is conducted in an effort to limit the computational and organization resources required. The ideal approach would require a fully-coupled high fidelity multi-component, multi-disciplinary analysis which would implicitly account for any intermediate variable statistical complexity. Such an analysis environment would be prohibitive to create, let alone execute.

A noteworthy contribution of the work discussed in Chapter 3 is that using the IPDF or MFOSM methods one can efficiently query the upstream analyses with as little as $n + 1$ evaluations and properly quantify the joint randomnes (i.e. statistical properties) of the upstream variables used as input to the component specific analyses. The implications of this interface between the component and the system will be the subject of subsequent sections of the demonstrated framework.

## *4.6 Local Statistical Space Characterization*

*Which variables should be modeled as random? Is statistical dependency present? How can one account for such statistical dependency?*

The objective of this step is to characterize the local parameter space, as depicted in Figure 47. This is an importance step to the posture taken within the current study as the local parameter space is then used as an input to the component reliability assessment. The conventional approach is to make several, usually strong, assumptions about the statistical behavior of the input crossing this boundary (reference Figure 46). Usually, because statistical knowledge of the local parameters is lacking or necessary computational resources are not available, these local input parameters are selected based on experience and assumed to be normally distributed and statistically independent. The mean and variance of each variable, required as input for its Normal distribution parameters, are at best quantified using a limited data set and usually assigned based on engineering judgement. However, statistical dependence between some or all of the input variables should be investigated as these physical variables

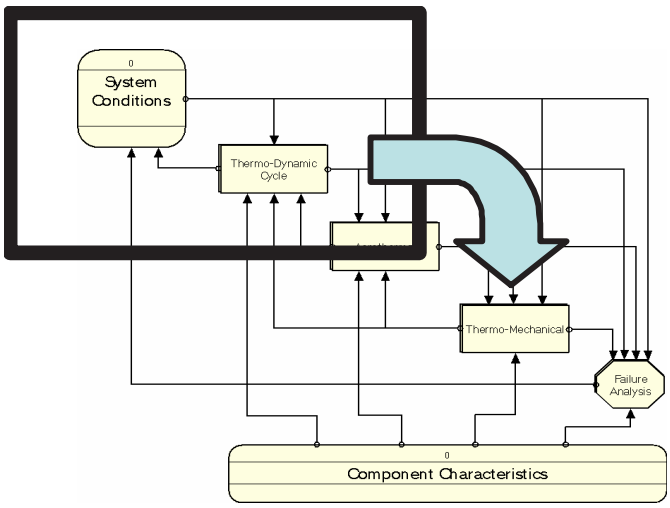


**Figure 47:** Influence of Upstream Analyses on Local Parameter Space

likely would share common upstream variables and dependency could greatly affect the probabilistic solution [93]. A potential alternative would be to expand the scope of the control volume to the point where there are few input variables that are easily predicted or even measurable. This point is usually the top-level system analysis where the input consists of operating and environmental parameters that are easier to measure and predict. However, general organizational limitations on human and computational resources usually impose a deterrent to this approach especially for probabilistic analyses which require numerous evaluations of the simulation space. A more efficient approach would be to limit the scope to that of the component analyses and capture critical statistical information driven by the upstream system behavior. Various methods, both new and old, are now prescribed that can capture this complex local parameter space. The various schemes are listed in Table 8. They include the newly developed sensitivity-based methods, IPDF (scheme 4) and MFOSM (scheme 3), and a combination of NPDF and either Monte Carlo analysis of applied Statistical Models (scheme 2) or direct Monte Carlo analysis of the physical model (scheme 1).

**Table 8:** Statistical Characterization Schemes.

| Scheme | JPDF | Statistics | Analysis | Evaluations (8 Vars) |
|---|---|---|---|---|
| 1 | NPDF | Monte Carlo | Physical Model | $> 1,000$ |
| 2 | NPDF | Monte Carlo | Metamodel | $2n^f + 2n + 1$ (84) |
| 3 | MFOSM | Covar. Approx. | Physical Model | $3n + 1$ (25) |
| 4 | IPDF | – | Physical Model | $3n + 1$ (25) |

### 4.6.1 Sensitivity-based Methods: MFOSM and IPDF

Sensitivity methods entail conducting a sensitivity analysis of each of the local input parameters as a function of the driving upstream system variables. This is accomplished by implementing a partial derivative of the functional relationship, $\mathbf{X}_i = f_i(\mathbf{X}_{global})$. For instance, the sensitivity of the $i^{th}$ intermediate variable, $\mathbf{X}_i$, would be computed by solving for the partial derivative with respect to each top-level system variable given by

$$\nabla \mathrm{X}_i = \left\{ \begin{array}{c} \frac{\partial f_i(\mathbf{X}_{global})}{\partial \mathbf{X}_{global_1}} \\ \frac{\partial f_i(\mathbf{X}_{global})}{\partial \mathbf{X}_{global_2}} \\ \vdots \\ \frac{\partial f_i(\mathbf{X}_{global})}{\partial \mathbf{X}_{global_n}} \end{array} \right\} \tag{76}$$

where $\mathbf{X}_{global_j}$ is the j$^{th}$ top-level global input to the system. However, in practice the functional relationship between the intermediate variables, $\mathbf{X}_i = f_i(\mathbf{X}_{global})$, may be unknown. Thus, a numerical implementation of equation (76) would be required. The finite difference approach [9] is recommended for a deterministic first-order first derivative approximation and would be of the following form

$$\frac{\partial \mathrm{X}_i}{\partial \mathrm{X}_{global_j}} \approx \frac{\triangle \mathrm{X}_i}{\triangle \mathbf{X}_{global_j}} \tag{77}$$

where $\triangle \mathbf{X}_{global_j}$ would be chosen based on the type of numerical differentiation to be performed. For example, for forward finite difference the perturbation of each input variable would be a small scalar or adder applied to each input variable. If statistical information, such as mean and standard deviation, is available on the input variables,

one might use the Taylor Series Forward Finite Difference (TSFFD) estimation [9] procedure which conducts the sensitivity analysis about the mean value of the input vector using a range of plus one standard deviation about each input variable given as

$$\triangle \mathbf{X}_{global_j} = (\mu_{X_{global_i}} + \sigma_{X_{global_i}})$$

and the intermediate variable vector evaluated at these maximum and minimum points of the input variable while the other variables in the input vector set to their mean value.

The baseline point at which the sensitivity measures are calculated should be taken at the expected value of the global input vector, $\overline{\mathbf{X}}_{global}$, thus yielding the expected value of the local (intermediate) input vector given as

$$E(\mathbf{X}_i) = \overline{\mathbf{X}}_i \approx f_i(\overline{\mathbf{X}}_{global}) \tag{78}$$

For the TSFFD method, because of the need to evaluate each response for both the high values for each input variable in addition to the mean value point, the number of function evaluations would be $n + 1$ where $n$ is the number of input variables.

Both the new methods MFOSM (scheme 3) and IPDF (scheme 4) use TSFFD but do so in two very different ways. For the MFOSM, the necessary covariance matrix is approximated using the formulae derived in Appendix A repeated here as

$$COV(X_i, X_j) \approx \sum_{l=1}^{n} \sum_{m=l}^{n} \left( \frac{\triangle \mathrm{X}_i}{\triangle \mathbf{X}_{global_l}} \right) \left( \frac{\triangle \mathrm{X}_j}{\triangle \mathbf{X}_{global_m}} \right) COV(\mathbf{X}_{global_l}, \mathbf{X}_{global_m}) \tag{79}$$

where $n$ is the number of global input variables. This formula uses a sensitivity derivative as input which can be provided by the TSFFD method. Also, notice that equation (79) reduces to the $VAR(X_i)$ when $COV(X_i, X_i)$ is sought. Assuming that the parameters follow a joint normal distribution then the necessary mean and covariance parameters are provided by the mean intermediate variable vector (equation 78) and covariance matrix, whose elements are given by equation (79). The resulting

joint normal probability density function is given as

$$f_{\mathbf{X}}(\mathbf{X}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} exp\left[-\frac{1}{2}(\mathbf{X}-\overline{\mathbf{X}})^T\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\overline{\mathbf{X}})\right] \tag{80}$$

where $T$ is the transpose operator and $\boldsymbol{\Sigma}$ is the covariance matrix whose $i,j^{th}$ element is $COV(X_i, X_j)$ [76]. Once the mean and covariance information is approximated using the sensitivity approach, the joint normal probability density function could then be employed as the stochastic characterization of the intermediate input.

The IPDF method utilizes the sensitivity derivative approximation in a different way. Instead of approximating response statistics directly, the sensitivity information is used to construct a multi-response Taylor series first-order, linear approximation of the response analysis. Provided that the joint distribution of the input parameters is known, the inverse solution (equation (52)) of the linearized system (equation (51)) would be solved for and equation (53) would be used to find the joint distribution of the local parameters. This process is not as straightforward as the MFOSM method is. However, IPDF is recommended when the global input is non-normal as the MFOSM is limited to joint normal input.

An additional step is required to implement the IPDF method when there are an unequal number of local and input parameters because of the matrix inversion limitation of the inverse solution. A symmetric coefficient matrix is required for matrix inversion to be performed. This limitation can be overcome easily for either of two possible scenarios. One scenario would be that there are more local parameters than there are global input parameters. In this case, extraneous variables could be added temporarily to allow the inverse solution to be found. When the joint probability distribution is to be solved for, the mean and variance values of these extraneous values can be set to zero and near zero, respectively. A more likely case is when there are more inputs than local parameters. Auxiliary equations can be created temporarily to allow for the inverse solution to be found. The resulting joint probability distribution function will contain these extraneous functions. This

is not a problem as the joint distribution of the pertinent local parameters can be solved by finding, through integration, the partial marginal distribution of the original distribution function (reference equation 37).

#### *4.6.1.1 Improved MFOSM and IPDF*

In their current form, both the MFOSM and IPDF methods share a common limitation. Since both are based on a first-order approximation, they are limited with respect to the joint response distribution that they can model. For instance, since MFOSM only provides the mean and covariance information, without further information on the actual response distributions one would have to assume the actual parameter marginal distributions. In this case, assuming a joint normal space is appropriate unless additional knowledge is available to suggest using another distribution. For the IPDF method, by the theory of a linear sum of normally distributed random variables joint normal input would result in a joint normal space being predicted. IPDF in its current linear-based form would predict non-normal response behavior only if the input distributions were non-normal. However, even with normal input the quadratic validation case of chapter 3 illustrated that the response distributions can be non-normal.

An improvement to these methods is now proposed to meet the needs of the current study. The improvement conceived can be either of the following optional steps: 1) Efficiently identify the underlying response distribution using small-sample statistical hypothesis testing, or 2) Use a second-order rather than a first-order approach within the MFSOM and IPDF methods. The second-order approach will be considered for future work as the covariance approximation formulae and IPDF integration solution would have to be re-derived for second-order information. Interestingly, the first approach can be undertaken with the already existing tools and information used within this study.

This approach involves using a small randomly generated sample of the function space to determine, through a statistical technique, the appropriate response distributions. The procedure for implementing this approach is to execute the analysis(es) for a small number of randomly generated cases. Choosing the number of cases is arbitrary. Statistically the chance of finding the proper distribution is increased with an increasing number of cases yet a minimum number of cases is strongly desired for efficiency as compared to alternative methods. As such, a rule of thumb of $2n$ random cases is recommended. These would be evaluated in addition to the original $n+1$ sensitivity cases for a total of $3n+1$ evaluations for either method.[2] Next, the A-D test statistic described in section 2.4.1 could then be utilized with the random case results to pinpoint the most appropriate marginal distribution for each local parameter. The unique non-normal distribution parameters would then have to be solved using the approximate covariance and mean values as input. Conceptually this improvement is quite unique in that the new joint probability methods, although linear in nature, can be used to estimate the parameters of what might be non-normal distributions, and therefore in effect predict non-linear, asymmetric random behavior.

In general, since only two parameters can be estimated for any of the responses using the MFOSM or IPDF approaches, this improvement step is limited to one or two-parameter probability distributions. Although limited to at most two-parameter distributions, many of the routinely used distributions have only two-parameters such as the often used Lognormal, Weibull, and Gamma distributions. The probability density function for each of these two-parameter distributions is given by Table 9. Since the MFOSM and IPDF methods would provide the mean and variance for each response, the unique non-normal distribution parameters must be calculated as a function of the response statistics. The formula relating the distribution parameters

[2]This rule of thumb value has additional relevance as the second order approach would require, as a minimum, $2n$ cases.

**Table 9:** Univariate Probability Density Functions [30].

| Name | Density Function |
|---|---|
| Lognormal[1] | $f(x;\mu_{ln},\sigma_{ln}) = \frac{1}{\sqrt{2\pi}\sigma_{ln}x} e^{-\frac{\left[log(x)-\mu_{ln}\right]^2}{2\sigma_{ln}^2}}$ |
| Weibull[1,2] | $f(x;\alpha,\beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-\frac{x}{\beta}^\alpha}$ |
| Gamma[1,2] | $f(x;\alpha,\beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}$ |

[1]Zero for negative values of x

[2]Where $\alpha$ and $\beta$ must satisfy $\alpha > 0$ and $\beta > 0$

**Table 10:** Distribution Parameter and Variable Statistics Equivalencies [30].

| Name | Density Function | Statistics |
|---|---|---|
| Lognormal | $f(x;\mu_{ln},\sigma_{ln})$ | $\mu = e^{\mu_{ln}+\frac{\sigma_{ln}^2}{2}}$<br>$\sigma^2 = e^{2\mu_{ln}+\sigma_{ln}^2} \cdot \left(e^{\sigma_{ln}^2-1} - 1\right)$ |
| Weibull | $f(x;\alpha,\beta)$ | $\mu = \beta\Gamma\left(1+\frac{1}{\alpha}\right)$<br>$\sigma^2 = \beta^2\left\{\Gamma\left(1+\frac{2}{\alpha}\right) - \left[\Gamma\left(1+\frac{1}{\alpha}\right)\right]^2\right\}$ |
| Gamma | $f(x;\alpha,\beta)$ | $\mu = \alpha\beta$<br>$\sigma^2 = \alpha\beta^2$ |

to the variable statistics for several non-normal distributions is given in Table 10.

An example of this distribution parameter solution sub-step is now given for the case where one must determine the parameters of a lognormal distribution function from the random variable mean and variance. Since a lognormal random variable is one in which the logarithm of the variable follows a normal distribution, the two distribution parameters will be represented using the following notation, $\mu_{ln}$ and $\sigma_{ln}$. These two parameters are the mean and standard deviation of the logarithm of the random variable and thus are not the actual variable mean and standard deviation. The lognormal distribution parameters are related to the variable mean and variance through the following implicit formulae [30]

$$\begin{aligned} \mu_{ln} &= log(\mu) - \sigma_{ln}^2/2 \\ \sigma^2 &= e^{2\mu_{ln}+\sigma_{ln}^2} \cdot \left(e^{\sigma_{ln}^2} - 1\right) \end{aligned} \tag{81}$$

where $\mu_{ln}$ and $\sigma_{ln}$ are the lognormal mean and standard deviation, respectively, and $\mu$ and $\sigma$ are the variable mean and standard deviation, respectively. The lognormal distribution parameters would then be solved iteratively using the variable mean and standard deviation as input. The variable mean and standard deviation would be provided by either the MFOSM or IPDF method. A complete quantification of the joint non-normal parameter space would then be produced by combining the marginal distribution identified for each local parameter with the parameter space covariance (correlations) computed either directly from MFOSM or through integration of the IPDF solution produced.

### 4.6.2 Simulation-based Method: NPDF

An alternative to the sensitivity-based schemes 3 and 4 would be to conduct a large sample simulation, Monte Carlo, analysis and use the NPDF method to construct the joint probability distribution. The NPDF method can be implemented using the results of a direct Monte Carlo analysis of the actual physical model (scheme 1) or using Monte Carlo analysis of an applied statistical model, i.e. metamodel, of the response space (scheme 2). A direct Monte Carlo analysis would require thousands of evaluations of the physical model and thus is only recommended for computationally inexpensive physical models. The metamodel approach is recommended if the functional forms of the responses are of a nonlinear nature which cannot be modeled using the first-order MFOSM or IPDF approach and yet also involves time-consuming analyses(is). The DOE sampling is the first major step within the Reponse Surface Method (RSM) where a Response Surface Equation (RSE) for each local input parameter is generated. Compared to the sensitivity method used for MFOSM and IPDF, the metamodel method usually requires many more evaluations of the upstream analyses, especially when considering quadratic or even higher order behavior. Yet, it can

yield a more accurate model of the complex local parameter deterministic space verifiable with several statistical measures for accuracy and validity. Both the first-order sensitivity and higher order metamodel methods are in fact methods of producing polynomial approximations to a response space of interest.

In the NPDF joint probability method, an empirical distribution function is required to provide the necessary joint probability information. The EDF is typically provided through large-sample Monte Carlo evaluations of the functional space. This empirical function would then be used as input to the A-D distribution identification procedure of section 2.4.1 to identify the appropriate marginal distribution function for each response variable. Necessary correlation information between the parameters could be calculated using a sample-based correlation analysis of the Monte Carlo response samples. Both of the two NPDF schemes, however, are inefficient compared to MFOSM or IPDF.

## *4.7 Parameter Space Reduction*

*Is there a subset of parameters that can be neglected during the component reliability analysis? Which contributing analyses can be omitted from parameterization? What component regions are of interest and how can they be identified?*

More often than not, a large and initially unmanageable set of input parameters are identified for a reliability analysis. The curse of dimensionality is an issue for deterministic and especially for non-deterministic analyses. Furthermore, for component reliability analyses involving FEA, each analysis requires considerable computational resources and time. Therefore, it is advantageous to perform a parameter reduction step, depicted in Figure 48, before conducting the component reliability assessment. As shown, this step involves ranking the parameters using one of many importance measures to be described herein. Since the preceding steps in the framework involved

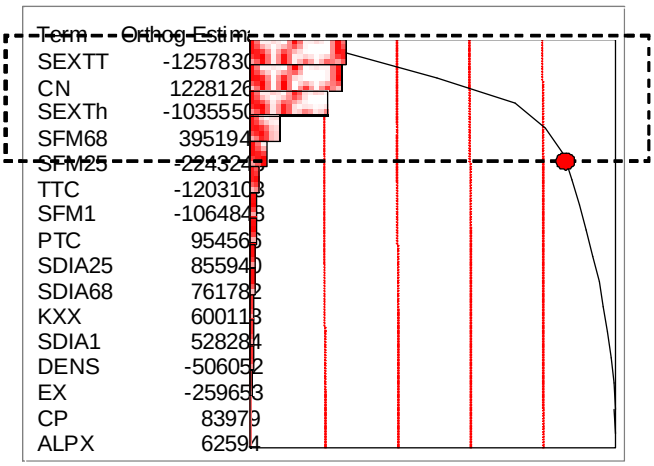


**Figure 48:** Parameter Space Reduction

creating an appropriate component modeling and simulation environment and quantifying the intermediate parameter stochastic space, the parameter reduction step can be guided with this information.

The reduction of parameters is typically accomplished using some sort of sensitivity measure that can be used to rank the importance of each variable on the variation of the response of interest. Existing parameter screening techniques that use deterministic or non-deterministic sensitivity measures are described. However, component reliability analyses can involve several failure responses in addition to a state of significant joint randomness. Therefore a multi-response parameter reduction technique is proposed.

### 4.7.1 Single Response Sensitivity Measures

#### *4.7.1.1 Deterministic Sensitivity Measure*

Deterministic sensitivity screening is usually conducted using approximate differential calculus. A partial derivative of the response function, $y(\mathbf{x}) = y(x_1, x_2, \ldots, x_n)$, versus each variable is calculated either analytically using differential calculus or numerical differentiation. The deterministic sensitivity parameter is

$$\frac{\partial y}{\partial x_i} \tag{82}$$

This is a commonly made calculation and is already described in earlier sections. For parameter screening, the sensitivity values for each response as a function of each input variable can be computed and used to rank order the parameters. Insignificant parameters can then quickly be identified.

#### *4.7.1.2 Non-deterministic Sensitivity Measure*

Since probability measures, such as reliability, are a function of the response variation induced by the input parameters. The deterministic sensitivity measure of equation (82) doesn't not consider the importance of each variable due to its statistical properties and thus is not a good screening measure. For instance, a less important variable in a deterministic sense might be relatively more important in a probabilistic sense should it have considerable variation. An additional nondeterministic consideration is whether or not the joint randomness, i.e. correlation, could affect the significance of each of the variables under consideration. Thus a non-deterministic sensitivity parameter is desired. Such a parameter is possible using the square correlation between the response parameter and each input parameter given as

$$\rho^2_{y_i,x_j} = \left( \frac{COV(y_i, x_j)}{\sigma_{y_i}\sigma_{x_j}} \right)^2 \tag{83}$$

where $COV\left(y_i, x_j\right)$ is the covariance between the the $y_i^{th}$ response and the $x_j^{th}$ input parameter. The covariance can be accurately approximated using the following covariance approximation formula (reference Appendix A)

$$COV\left(y_i, x_j\right) \approx \sum_{k=1}^{n} \frac{\partial y_i}{\partial x_k} COV\left(x_k, x_j\right) \tag{84}$$

### 4.7.2 Single Response Deterministic Pareto Screening

Single-response pareto screening, also known as a screening DOE, is often used in statistics as a parameter screening step prior to the creation of a higher order applied statistics model. It is applied individually to each response. A cursory overview of the

pareto screening method is now given. Usually a fractional factorial, two-level DOE involving an initially large input parameter set is constructed and executed for each response. A linear statistical model and subsequent ANalysis Of Variance (ANOVA) is conducted so as to compute the coefficient estimate for each term in the model and in turn determine a relative measure of the importance of each input parameter. Starting with the most important parameter first, more and more parameters can be considered until roughly 80% of the variation of the response is accounted for. This process would have to be repeated for each response until all of the significant input parameters have been identified. This method is analogous to the deterministic sensitivity approach in that it is essentially a sensitivity measure but differs in how the individual input parameter values are created for each case as well as the actual formula used to rank the input parameters. But, it also shares the same limitation as the deterministic sensitivity approach in that statistical variation, particularly correlation, of the input parameters is not considered. The non-deterministic sensitivity method previously described can consider complex statistical behavior of the input, but, as with the pareto and deterministic sensitivity approaches, it is a single response method.

### 4.7.3 Multi-Response Non-Deterministic Screening Method

The previously discussed non-deterministic sensitivity-based reduction method allows for the first order second moment information of the input (mean and variance vectors) to be considered. If the probabilistic relationship between a vector of component failure responses and a vector of input variables is sought, then a different measure is required. Fortunately, Canonical Correlation Analysis (CCA) [47] is a technique that can be used for this purpose. CCA, however, is a sample-based approach requiring statistical samples of the various parameters. But, using the covariance approximation formulae derived for this study, analytically generated input to CCA using the system analysis structure is now possible and is dubbed *Approximate Canonical Correlation*

*Analysis (ACCA).* The background of canonical correlation is now given.

Canonical correlation is a statistical measure defined as the statistical linear correlation between two *sets* of variables. Canonical correlation was first introduced by Hotelling in 1936 [47] as a means of calculating the amount of statistical linear relationship between two vectors, or sets of variables. Where traditional correlation is concerned with the statistical linear dependence between two variables $x_i$ and $x_j$, and multiple correlation is concerned with the correlation between one response, $y$, and several $x$'s; canonical correlation is an extension of this concept that can be used to compute the correlation between a vector of responses, $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, and a vector of input variables, $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$.

The usefulness of this measure will be shown in being able to compute the degree of correlation between the vector of failure responses and prospective input vector subsets that represent various attributes of the complex system such as the contributing analysis or a class of input variables (i.e. thermal material properties). Thus, a means of computing the sensitivity of the failure analysis to an entire contributing analysis (see Figure 46) is possible which would permit the relative ranking of contributing analyses according to their influence on the variation of the component reliability.

Canonical correlations can be computed using the correlation coefficient matrix between the responses, $R_{yy}$, the responses and the inputs, $R_{yx}$, and the correlation matrix of the input, $R_{xx}$. The resulting *composite* correlation matrix would be given as

$$\left[\begin{array}{c|c} R(\mathbf{y},\mathbf{y})_{n\times n} & R(\mathbf{y},\mathbf{x})_{n\times m} \\ \hline R(\mathbf{x},\mathbf{y})_{m\times n} & R(\mathbf{x},\mathbf{x})_{m\times m} \end{array}\right]_{(n+m)\times(n+m)}$$

where $n$ and $m$ are the number of response and input parameters, respectively. Further, where canonical correlation has been applied to statistics of sample data, it is applied here in a new way using analytically derived statistics. *The covariation approximation technique developed in this study are used to produce the required correlation*

*structure.*

Once the composite correlation matrix is constructed, whether from the covariance approximation technique or from a conventional sample-based algorithm, the objective then is to use the correlation structure to find the underlying statistical sensitivity and correlation between a vector of responses and a vector of input variables. This correlation coefficient, or canonical correlation, is determined by solving the following two simultaneous equations

$$\left(R_{yy}^{-1}R_{yx}R_{xx}^{-1}R_{xy} - \mathbf{r}^2 I\right) f = 0$$
$$\left(R_{xx}^{-1}R_{xy}R_{yy}^{-1}R_{yx} - \mathbf{r}^2 I\right) g = 0$$

where $I$ is the identity matrix, $f$ and $g$ are the eigenvectors, and $\mathbf{r}$ is the vector of canonical correlations for this vector pair. The canonical correlations can be calculated by solving for the eigenvalues of the following characteristic equations

$$\left|R_{yy}^{-1}R_{yx}R_{xx}^{-1}R_{xy} - \mathbf{r}^2 I\right| = 0$$
$$\left|R_{xx}^{-1}R_{xy}R_{yy}^{-1}R_{yx} - \mathbf{r}^2 I\right| = 0$$

where $s = min(n, m)$ and is the number of eigenvalues, $r_1^2, r_2^2, \ldots, r_s^2$. The $s$ eigenvalues are known as the square canonical correlations. Although several canonical correlation values are produced, the value reported is the maximum square canonical correlation, $r^2 = max\{r_1^2, r_2^2, \ldots, r_s^2\}$ and is used as the measure of association between the two vectors.

This measure of association is highly useful as it is analogous to the statistical parameter called the coefficient of determination which is used to quantify the fitness of a single linear function of multiple variables. The difference between the two is that all of the responses and all of the input variables are considered simultaneously with the canonical correlation calculation. As with the traditional correlation coefficient, although the canonical correlations can be either positive or negative the range is between $-1$ and 1 and the square of the canonical correlations will always vary from 0 to unity. Because of these properties, Canonical Correlation Analysis (CCA) is a very

useful parameter space reduction technique for a multi-response space and is ideally suited for this study in combination with the novel covariance approximation technique. Approximate Canonical Correlation Analysis (ACCA) is CCA implemented using the covariance (correlation) approximation formulae derived within this study.

Within the formulated framework, ACCA is specifically used to reduce the parameter space in the following manner. After the complete composite correlation matrix between the failure responses is computed using either the covariance approximation or inverse transformation techniques, several canonical correlations across several hierarchical levels are to be evaluated. For example, maximum square canonical correlation values can be computed for the response vector and the vector of intermediate parameters from each contributing analysis. In this posture, one could rank order the contributing analyses using this metric and determine which of these are the most important contributors to the entire vector of failure responses and even neglect analyses that are determined to be insignificant. Or, maximum square canonical correlation values can be computed between the vector of failure responses and the various vectors that are input to a contributing analysis but represent a different class of input. For instance, one might be interested in the rank order of importance of subsets of input parameters to the material property analysis representing basic mechanical versus thermal versus failure categories. Such information would be invaluable for an organization in determining where to focus efforts on development and quality control activities. This method can also be used to greatly improve the optimization of a system with multiple responses as the canonical correlation values could be used to weight the search direction and even step length of the process. *Therefore, ACCA is recommended, in general, for parameter space reduction for problems involving multiple responses in the presence of uncertainty, and, specifically, for component reliability activities.*

## 4.8 Probabilistic Failure Assessment

*What probabilistic methods are available and when should you use them? Which combinations of upstream joint probability models and structural reliability methods should be considered?*

The preceding steps in the newly developed framework are used to greatly improve the input information and posture before the last step, probabilistic failure assessment, is conducted. One would then use the local parameter space model created using one of the schemes described in section 4.6 as input to the probabilistic assessment. However, several successful probabilistic reliability methods already exist, as discussed in section 2.3. The major categories of available probabilistic analyses include simulation-based, analytical most probable point methods, and hybrid methods such as simulation combined with an applied statistical model. If explicit, short running analyses or models are available for all of the elements of the component failure environment then a full simulation approach is recommended for the probabilistic analysis. Should this be the case, the NPDF approach for characterizing the local parameter space is recommended. However, most upstream, system analysis as well as component-specific analyses are computationally intensive. In this scenario, the efficient MFOSM or IPDF methods are recommended as a means of quantifying the local parameter space and either the hybrid simulation-applied statistical model or most probable point probabilistic methods are recommended for the component probabilistic failure analysis. To use the MFOSM or IPDF methods as input to the component reliability assessment one would use an appropriate probability space transformation technique from those listed in section 3.5.2.

# CHAPTER V

# IMPLEMENTATION AND RESULTS

An application of the previously described framework is now given for a relevant industry problem. The problem is to efficiently, yet accurately assess the reliability of a gas turbine blade for a simplified cruise operating condition. This problem is of great importance to the aerospace and power generation industries as hot gas path turbine components exhibit considerable scatter in their useful life which can only partially be explained through inherent variation of the components properties. The remaining variation is attributed to the interaction between the component and the system within which it operates.

Hot gas path components are exposed to severe and complex boundary conditions as well as numerous additional sources of variation during their design, production, and operational life cycle. Thus, a complex environment is necessary which can permit the integration of primary sources of component failure uncertainty. A suitable multi-physics environment has been created to automate the bulk creep life assessment of the aircraft gas turbine airfoil as a function of material and system-induced variation and enable the execution of the developed framework discussed in Chapter 4.

## *5.1 Component Identification*

The safe, reliable operation of the entire aircraft is highly dependent on the propulsion system utilized. One of the most critical components affecting the safety and reliability of turbine engines is the turbine blade. Failure of a turbine airfoil could cause a high-energy part to be released from the rotor system destroying the entire

engine through a cascade of subsequent events. Such a situation could cause a catastrophic condition for the entire vehicle. Therefore, the propulsion system reliability is determined, within this study, by conducting a reliability assessment of a single turbine airfoil[1].

## *5.2 Decomposition*

The initial step taken during decomposition is to assess the functional characteristics of the component relative to the system-level safety and reliability goals. The turbine blade functionality can be illustrated using Figure 49. Here, a control volume is drawn which represents a boundary between the physical component perimeter and any medium which work is imposed to or from the component for it to complete its function. In this case, high energy flow leaving the combustor strikes the turbine blade and by virtue of the aerodynamic shape of the blade some of this energy is converted into a resultant lift vector on the blade which results in a torque applied to the turbine rotor. Since the airfoil is exposed to exceedingly high temperatures, coolant flow at a much lower temperature is extracted from the compressor is pumped through the internal passages of the blade to assist in maintaining the blade metal temperature. The external fluid imposes work on the blade while the internal flow extracts heat. This results in the conversion of some of the core flow energy to shaft power to drive the compressor as well as lower energy core flow and higher energy coolant flow. Another implied, yet no less important, function is to maintain the safety of the system. A failure of the component in terms of separation would likely result in a cascading and perhaps even a catastrophic failure of the system. Thus, the functional purpose of the turbine blade is to safely convert some of the high energy core flow into shaft power to drive the compressor.

[1]Should this apriori information be unavailable, especially under a design or even concept change, then the more involved yet more complete approach prescribed in section 4.3 is recommended.

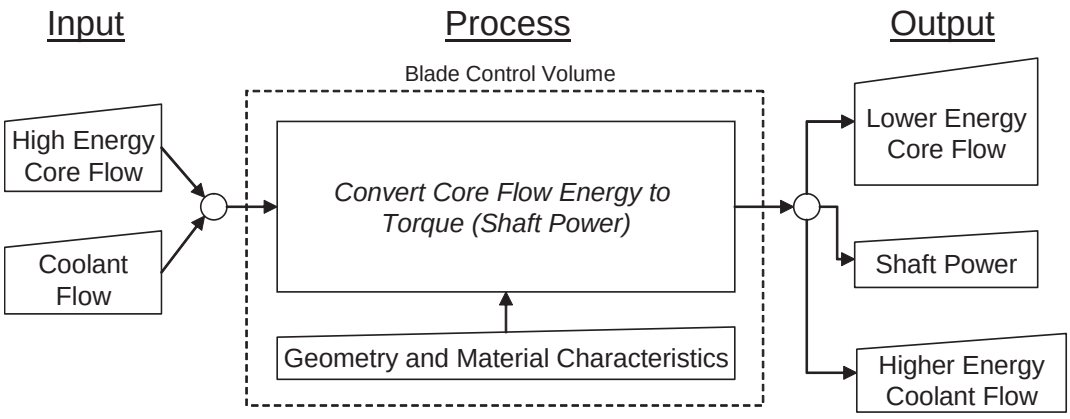


**Figure 49:** Blade Functional Relationship Diagram

Now that the driving functions of the blade have been identified the failure modes that could lead to a malfunction are to be determined. There are numerous potential failure modes of a second stage turbine blade. However, experience and judgement can be used at this point in leu of an actual component model to identify the potential failure modes. Three failure modes[2] are considered for this $2^{nd}$ stage turbine blade based on a previously completed study of a similar component under similar operating conditions [95][94][96]. They are *creep rupture*, because of the anticipated high operating temperature and stress; *fatigue* due to the high cyclic stress induced by the rotational speed of the rotor; and material strength *overstress* which is also driven by the rotor speed. These failure mechanisms are discussed further in the synthesis step described in section 5.3. For now, cursory knowledge of these failure mechanisms provides a useful insight to the relationship between the failure modes and the parameters identified through the functional relationship exercise (see Figure 49). This relationship is summarized in Table 11. Each of the three failure modes can prevent either of the two functions, maintain safety and extract/convert flow energy, so these boxes are fully populated. Also, creep rupture is known to be a

[2]A complete list of potential gas turbine blade failure modes is included in Figure 5. Of these, corrosion and oxidation are common modes but are usually found in $1^{st}$ stage blades rather than latter stages.

**Table 11:** Blade Failure Mode Identification.

| | **Function** | | **Parameters** | | | | |
|---|---|---|---|---|---|---|---|
| | | | **Mass** | | **Energy** | **Characteristics** | |
| **Failure Mode** | Safety | Extract Energy | Core Flow | Coolant Flow | Rotor Speed | Geometry | Material |
| Creep | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Fatigue | ✓ | ✓ | | | ✓ | ✓ | ✓ |
| Overstress | ✓ | ✓ | | | ✓ | ✓ | ✓ |

direct function of temperature, stress, and of course material properties so all of the parameters are considered important for this failure mode. The fatigue and overstress failure modes are dependent mainly on the stress and material properties of the the component. Since the stress is a strong direct function of the rotor speed and geometry, these parameters are highlighted as well. Of course, one could argue that the thermal parameters (flow parameters) also would be significant. However, this is just an information management step necessary as input to the following synthesis step described in section 5.3. A more quantitative and results driven assessment will follow during the parameter reduction step described in section 5.5.

The result of the decomposition step is an initial characterization of the failure modes, necessary parameters, and of course an idea of what analyses are required. For instance, all three modes require knowledge of the state of stress of the component so a mechanical analysis is required. Also, the creep rupture time analysis as well as the required material model both require a prediction of the metal temperature distribution and thus a thermal solid analysis is required. Also, the parameters identified in Table 11 are further classified across several categories as shown in Table 12 which provide additional insight to the conditions leading to the synthesis of the component analysis approach.

**Table 12:** Parameter Classification.

| | | Parameters | | | | |
|---|---|---|---|---|---|---|
| | | Mass | | Energy | Characteristics | |
| **Category** | **Descriptor** | Core Flow | Coolant Flow | Rotor Speed | Geometry | Material |
| **Behavior** | Constant | | | | ✓ | ✓ |
| | Temporal | ✓ | ✓ | ✓ | | |
| | Spatial | ✓ | ✓ | | ✓ | ✓ |
| **Variation** | Component | | | | ✓ | ✓ |
| | System | ✓ | ✓ | ✓ | | |
| **Control** | Design | ✓ | ✓ | ✓ | ✓ | |
| | Noise | ✓ | ✓ | ✓ | ✓ | ✓ |

## *5.3 Synthesis*

Now that the failure posture of the blade has been assessed, an appropriate failure modelling and simulation environment is sought. Within this step, the actual analyses, their complementary assumptions, and flow of information between the system-level, intermediate, and failure analyses are defined and a baseline analysis performed. Due to the repetitive nature of the probabilistic analyses to be subsequently executed, an environment described in section 5.3.6 was created to automate the entire analysis structure thereby maximizing the efficiency of the subsequent steps and minimizing the chance of user induced artificial error.

### 5.3.1 System Model: Aircraft Jet Engine

The propulsion system selected for this study is a generic separate flow turbofan engine typical of those that power medium sized transport aircraft such as the B737 aircraft. Overall pressure and bypass ratios of the propulsion system are roughly 30:1 and 6:1, respectively. The maximum thrust at sea-level static conditions is around 20,000 $lb_f$. The vehicle mission used in this study is typical of a B737 aircraft; although, it has been simplified to facilitate the demonstration of the proposed

method. Only the cruise condition is considered for this study. The take-off and landing segments are modelled as discontinuous jumps in engine rotor speed between the shut-down and cruise segments of the operating profile. The cruise segment occurs at an altitude of 35,000 feet and a speed of Mach 0.745 as expected for this type of aircraft. The mission parameters identified for this study are the cruise altitude and Mach number, and the change in ambient temperature.

A $2^{nd}$ law thermo-dynamic cycle analysis is used to represent the jet engine and provide the system-driven aerothermal and mechanical input necessary for the subsequent component specific analyses. A baseline NEPP thermodynamic cycle model of the turbofan engine was utilized to compute the required 1-D steady-state engine station and component properties [4], [2]. The thermodynamic cycle parameters identified for this study are given in Table 13. They include three operational parameters representing the cruise flight condition, altitude, Mach, and ambient temperature change, $\Delta T_a$, as well as the efficiency of each of the major cycle components. The required output provided by the cycle analysis is the high-pressure rotor speed, RN, and turbine entrance and compressor exit temperatures, which are assigned as the blade coolant and external gas flow temperatures, $T_c$ and $T_g$, respectively[3]. These cycle responses are required as intermediate input for the subsequent component-specific analyses.

### 5.3.2 Solid Model: Thermo-Mechanical Finite Element Analysis

The 2nd stage turbine bucket is a complex 3-D part. To determine the complex state of solid temperature, stress, and strain, a 3-D finite element mesh, shown in Figure 50, was utilized. The finite element mesh consists of around 92,000 nodes and 75,000 elements. The calculation of the metal temperatures and mechanical stress

[3]Actually, the coolant and external gas temperatures will likely be different, in an absolute sense, from the cycle parameters used. However, these values are used as scalar modifiers during subsequent blade analyses and therefore only the relative change is required. Relative change in these sets of parameters should be consistent.

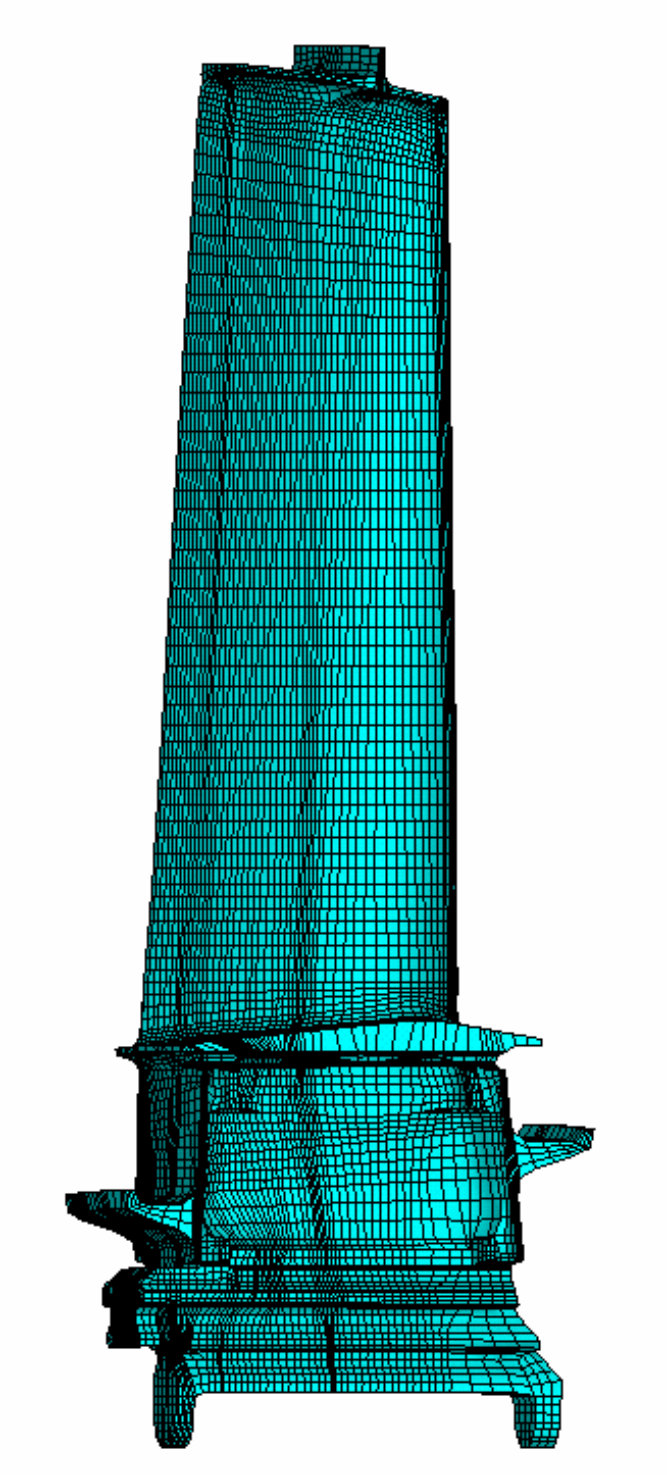

**Figure 50:** Blade Finite Element Mesh

**Table 13:** Engine Cycle Parameters.

| Parameter | Description | Class | Units | Mean | $\sigma$ | Distribution |
|---|---|---|---|---|---|---|
| Altitude | – | Control | feet | 35,000 | 1750 | Normal |
| Mach | Flight Speed | Control | – | 0.745 | 0.03725 | Normal |
| $\Delta T_a$ | Ambient Temp. Change | Noise | $^\circ F$ | 0 | 15 | Normal |
| $\eta_f$ | Fan Efficiency | Constant | – | 0.885 | – | – |
| $\eta_{lpc}$ | LPC Efficiency | Constant | – | 0.900 | – | – |
| $\eta_{hpc}$ | HPC Efficiency | Constant | – | 0.870 | – | – |
| $\eta_{hpt}$ | HPT Efficiency | Constant | – | 0.930 | – | – |
| $\eta_{lpt}$ | LPT Efficiency | Constant | – | 0.935 | – | – |
| $T_c$ | Coolant Temp. | Response | $^\circ F$ | – | – | – |
| $T_g$ | Core Temp. | Response | $^\circ F$ | – | – | – |
| RN | Rotor Speed | Response | rpm | – | – | – |

LPC low pressure compressor, HPC high pressure compressor
HPT high pressure turbine, LPT low pressure turbine

requires a thermo-mechanical solid analysis. This analysis is conducted using ANSYS 5.7 FEA software. The baseline finite element model consists of the finite element mesh, shown in Figure 50, thermal boundary conditions in the form of near-surface gas temperatures and heat transfer coefficient fields, shown in Figure 51, mechanical boundary conditions in the form of the solid temperature distribution[4], inertial load in the form of a rotor speed specification, and displacement boundary conditions, shown in Figure 52 as well as temperature dependent material properties. See section 5.3.3 for a description of the material properties used in this study. The thermo-mechanical boundary conditions and finite element mesh were generated originally for a $2^{nd}$ stage turbine blade for a land-based gas turbine operating at its base power mode [95][94][96], but scaled to meet the approximate size and boundary condition values for the jet engine operating at the nominal cruise condition section as described in section 5.3.1.

Although the thermal and mechanical analyses are in reality coupled, this study

[4]Solid temperature distribution is not shown in Figure 51.

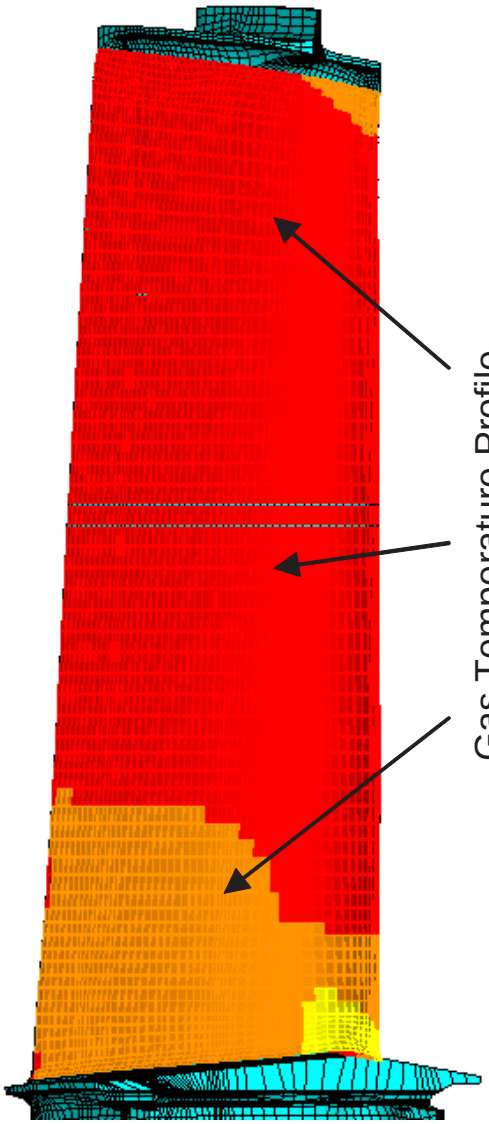


**Figure 51:** Thermal FEM Boundary Conditions

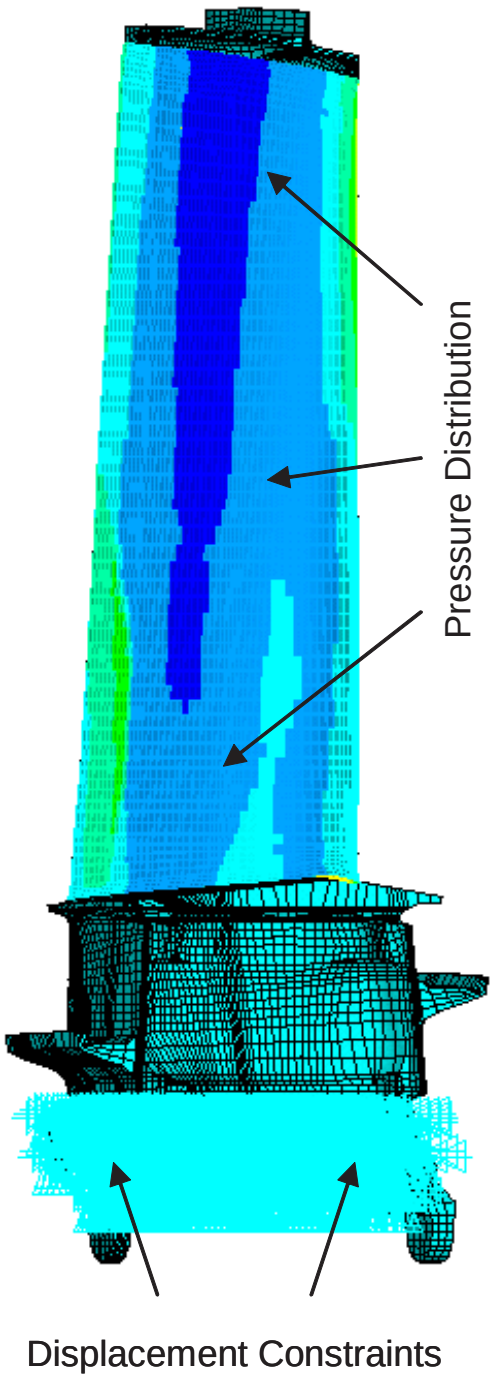


**Figure 52:** Mechanical FEM Boundary Conditions

models them as uncoupled to maintain an acceptable computational run time. Therefore, a converged thermal analysis is found and the resulting solid temperature solution provided as a boundary condition input for the stress (mechanical) analysis. The result of the thermal solid analysis during each iteration as the thermal solution converges is a completely stabilized metal temperature distribution which is provided as input to the material analysis. The material analysis in turn provides an updated material thermal property matrix to the thermal solid analysis and the process repeated. Consequently, a coupling is created through this dual-analysis interaction resulting in a non-linear FEA solution. Once the converged thermal solid solution is found, a full stress and strain distribution is determined through the mechanical analysis using the rotor speed and pressure distribution input from the cycle analysis, the metal temperature distribution from the thermal analysis, and the mechanical property matrix from the material analysis. The thermal and solid solutions are then provided as input to the failure mode analysis.

### 5.3.3 Material Model: Nimonic80A

The blade material selected for this study is the wrought, age-hardenable, nickel-based alloy Nimonic80A. Nimonic80A is in the older class of turbine blade materials. Even so it can still withstand continuous metal temperatures of up to $1500°F$. Several temperature dependent properties of Nimonic80A showing this resistance to temperature are illustrated in Figure 53. This material is comprised of around 70% nickel, around 20% chromium for increased oxidation resistance, around 2.3%, 1.4%, and 0.1% carbon, respectively, for added strength, as well as other elements such as Iron (3.0%) and cobalt (2.0%). Room temperature physical, mechanical, strength, creep rupture, and fatigue properties used in this study are listed in Table 14. Values for these parameters for temperatures ranging from room temperature ($68°F$) up to $1800°F$ were used as input to the various analyses and are provided in reference [1]. In addition to

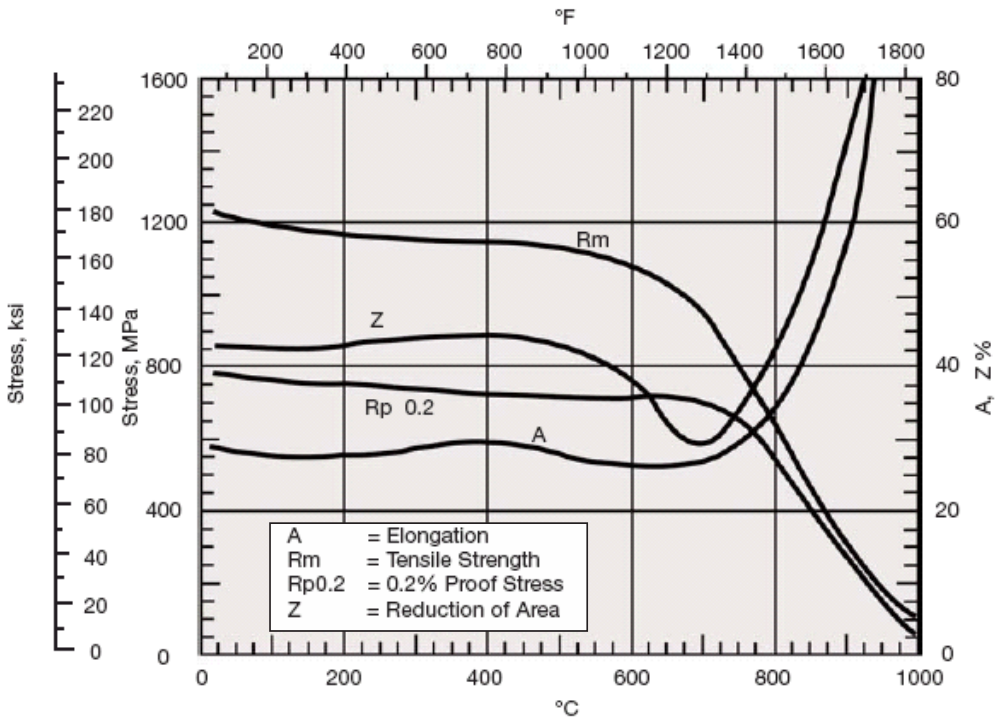


**Figure 53:** Temperature Dependent Strength Properties of Nimonic80A (Reproduced from [1])

the room temperature values, representative statistical properties of these properties are given in Table 14. These statistical properties are representative in that they were compiled from several references, listed at the bottom of Table 14, where the statistical variation of these properties were measured. The statistical properties were determined using data sets of the individual material properties evaluated at room temperature. Since statistical variation for higher temperatures was not available, it is assumed that the variation at any point along the temperature dependent material property curve has the same statistical variation as the room temperature variation. An illustration of this assumption is shown in Figure 54, where the room temperature variation is assumed constant along the entire material property curve. While the mean point may change along this curve, the standard deviation, or variance, remains the same at every point.

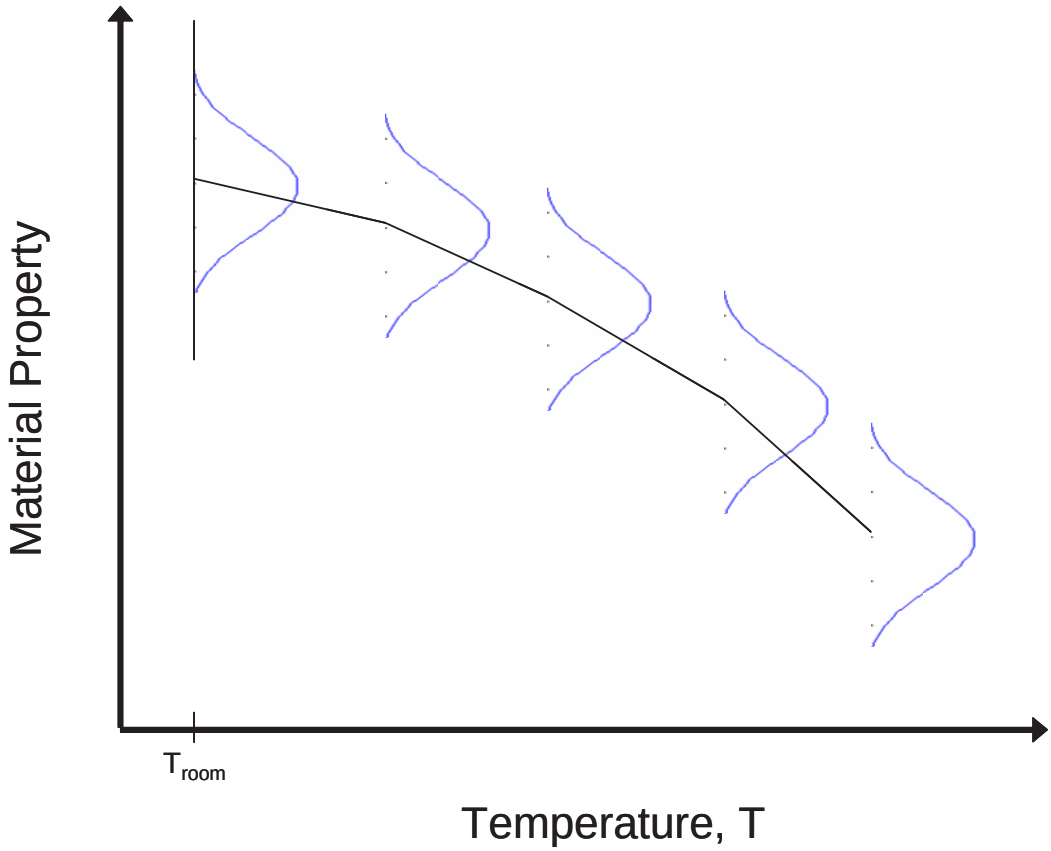


**Figure 54:** Constant Statistical Variance of Material Properties

**Table 14:** Blade Material Parameters (Nimonic80A).

| Parameter | Description | Units | $\mu$ | $\sigma$ | Distribution |
|---|---|---|---|---|---|
| EX | Elastic Modulus | psi | 33E6 | 1.65E6 | Normal[1] |
| NUXY | Poisson's Ratio | – | 0.3 | – | Constant |
| KXX | Thermal Conduct. | $\frac{Btu \cdot in}{sec \cdot ft^2 \cdot ^\circ F}$ | 1.4898E-4 | 2.998E-6 | Normal[2] |
| ALPX | Thermal Expansion | $^\circ F \cdot 10^{-6}$ | 7.05̄E-6 | 1.41̄E-7 | Normal[2] |
| CP | Specific Heat | $\frac{Btu}{lb \cdot ^\circ F}$ | 41.3 | – | Constant |
| DENS | Density | $\frac{lb}{g \cdot in^3}$ | 7.667E-4 | 1.533E-5 | Normal[1] |
| CN | Creep Variation | – | 0 | 1 | Normal[2] |
| SY | Yield Strength | ksi | 112 | 4.5 | Normal[3] |
| SU | Ultimate Strength | ksi | 180 | 7.2 | Normal[3] |
| SFP | Fatigue Constant | ksi | 327.05 | 6.2 | Normal[4] |
| B | Fatigue Exponent | – | -0.117 | 0.01053 | Normal[4] |

[1]Sues et al. [87]; [2]Wallace and Mavris [93][94]; [3]Wu, Y.-T. [100]
[4]Pascual, F.G. and Meeker,W.Q. [73]

### 5.3.4 Failure Models: Creep Rupture, Fatigue, and Overstress

Before the failure models can be utilized, a postprocessing mid-step of the mechanical analysis result must be undertaken. The output of the mechanical FEA solution is a complex, 3-dimensional state of stress at each element of the turbine blade model. However, the failure functions used in this study are based on empirical specimen data under 1-dimensional stress. Therefore, the complex state of stress must be converted to an equivalent 1-dimensional value. The state of stress of the turbine blade is modelled in this study as cyclical between the takeoff and cruise mission segments, yet remains at an elevated stress during the cruise condition. Assuming that the cyclic stress amplitude is constant and in-phase in all directions locally at each element, then the Von-Mises Octahedral shear stress criteria can be used to compute a 1-dimensional equivalent stress range given as

$$\triangle\overline{\sigma} = \frac{1}{\sqrt{2}}\sqrt{\left(\triangle\sigma_{xx} - \triangle\sigma_{yy}\right)^2 + \left(\triangle\sigma_{yy} - \triangle\sigma_{zz}\right)^2 + \left(\triangle\sigma_{xx} - \triangle\sigma_{zz}\right)^2 + 6\left(\sum_{i\neq j}\triangle\tau_{ij}^2\right)} \tag{85}$$

where $\sigma_{ii}$ is the normal stress in the $i^{th}$ direction, and $\tau_{ij}$ is the $ij$ shear stress. Since the equivalent mean stress is simply the sum of the mean of each normal stress component, then the mean, maximum, and minimum stress can be found easily. The maximum equivalent stress is used in the creep rupture calculation while the mean and stress amplitude, given as $\overline{\sigma}_m = \frac{1}{3}\sum_{i=1}^{3}\sigma_{ii,mean}$ and $\overline{\sigma}_a = \frac{\overline{\triangle\sigma}}{2}$, respectively, are used in the fatigue failure calculation. The mean and amplitude of stress are required for the fatigue failure calculation because the fatigue life data used in this study, as with most fatigue life data, is generated based on zero mean stress conditions where the cyclic stress is completely reversed. Thus, an equivalent completely reversed stress must be computed from the equivalent 1-D stress of equation (85) and is given as

$$\sigma_{ar} = \frac{\overline{\sigma}_a}{1 - \frac{\overline{\sigma}_m}{\sigma_u}} \tag{86}$$

where $\overline{\sigma}_m$ is the equivalent 1-D mean stress and $\sigma_u$ is the ultimate tensile strength of the material. The quantity given by equation (86) may be used directly with completely reversed stress versus life test data to find the predicted fatigue life of the element.

The overstress failure model used in this study is simply the ratio between the equivalent 1-D stress and the ultimate stress of the material. Thus, whenever this ratio becomes greater than 1, then failure occurs. However, if the material is chosen properly the ultimate strength of the material should never be breached and so a smaller overstress value is usually used as a failure criteria, such as 0.8. This failure mode is easily computed as the 1-D stress is produced by the mechanical analysis and the ultimate stress is determined from the material input matrix as a function of the temperature of the element.

The material creep rupture life model used in this study is the Orr-Sherby-Dorn (OSD) three-term function based on the Arrhenius activation energy [72]. The OSD creep life function gives the elapsed time when the material reaches a specified creep strain limit given as

$$t_{creep} = e^{\frac{1}{A+B\log(\overline{\sigma})}+\frac{\frac{-Q}{R}}{T+459.67}+CN\cdot SE} \tag{87}$$

where $\overline{\sigma}$ is the equivalent stress (resolved during the cruise operating condition), $T$ is the temperature, and the parameters A, B, and -Q/R are empirically determined creep constants corresponding to the time at which the accrual of a pre-defined amount of creep strain is reached. The third term in the exponential represents the statistical variation of the creep rupture life, where $CN$ is a standard normal variable and $SE$ is a constant also determined empirically from test data. Several tests at various levels of stress and temperature must be performed to determine the creep-strain limit constants. They are unique both to the type of material used as well as the creep strain limit specified. Using equation (87) the creep life at each element can be determined as a function of several upstream variables such as bucket external

and internal heat transfer boundary conditions, material properties, and geometry. Additional material characterization information for creep life modeling of nickel-based super alloys is given by Daleo et al. [23], [24].

Fatigue failure, is more involved. Assuming that linear-elastic conditions exist, the straightforward stress-life, S-N, approach [12] can be utilized to estimate the number of cycles to failure given completely reversed, constant stress cycle amplitude. The following relationship is commonly used in the stress-life approach

$$\sigma_{ar} = \sigma_f^{'}(2N_f)^b \tag{88}$$

where $\sigma_f^{'}$ is the fatigue strength coefficient, $N_f$ is the cycles to failure, and $b$ is the fatigue exponent. The two fatigue constants, $\sigma_f^{'}$ and $b$, are found by fitting the following two constants to stress versus life results from coupon size test specimens, $C_1 = 2^b\sigma_f^{'}$ and $C_2 = b$. The fatigue life for a given level of stress would then be computed using

$$N_f = 0.5\left(\frac{\sigma_{ar}}{\sigma_f^{'}}\right)^b \tag{89}$$

where $\sigma_{ar}$ is found using equation (86) when non-zero mean stress is present. This approach has been successfully applied to the case of tensile fatigue loading of ductile metals [33].

At this point, it is necessary to make another simplifying assumption with regards to the fatigue analysis. The Morrow fatigue function assumes that the stress cycle amplitude is constant. Consequently, the stress cycle generated within the context of this problem is defined as the centrifugal stress cycling between the zero stress condition before starting the engine and the stress state at the cruise condition. Realistically, the stress-state is highly complex in a temporal sense as the engine operating conditions vary greatly over the mission profile of the vehicle. Further, material property and geometry non-linearity and perhaps even temporal randomness may also exist, which further complicate the analysis. Several experts within the area

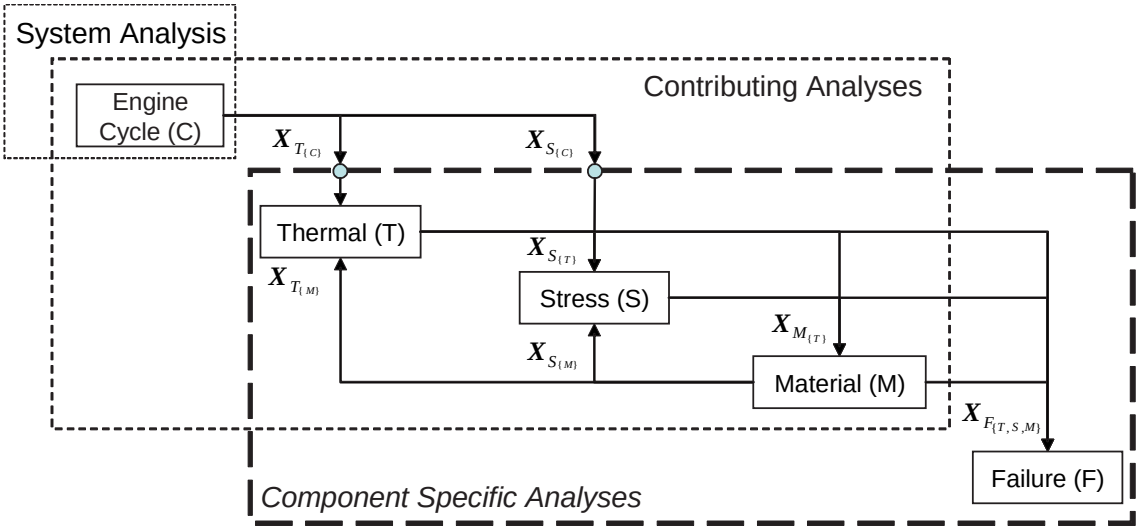


**Figure 55:** Blade Multi-Disciplinary Analysis Structure

of mechanics of materials have focused considerable effort towards improving the prediction of the complex stress-strain state as well as developing models to account for the complex interaction between the various high temperature failure mechanisms found in turbine blade applications. *Consequently, the failure functions considered are relatively straightforward in nature and are assumed to be independent of each other.*

### 5.3.5 Baseline Failure Analysis Results

Up to this point, all of the necessary input and boundary conditions have been described and specified for the multi-response failure analysis of the $2^{nd}$ stage turbine bucket. Expected values for each of the random variables are then chosen for the baseline case, and the entire analysis executed to determine the baseline failure solution. This process is depicted in Figure 55. First, system-level input (see Table 13) is provided to the system analysis module where the thermodynamic engine cycle analysis is run to determine the core flow gas temperature,$X_{T_{C1}}$, cooling flow temperature, $X_{T_{C2}}$, and turbine rotor speed. The turbine rotor speed is input to both the thermal and stress analysis modules denoted by $X_{T_{C3}}$ and $X_{S_{C3}}$, respectively.

The thermal analysis requires input from the cycle model, namely the core flow and coolant flow temperatures ($X_{T_{C1}}$, $X_{T_{C2}}$) and rotor speed[5]($X_{T_{C3}}$), as well as from the material model which provides the thermal material properties ($X_{T_{M1}}$). The blade temperature distribution is determined during this step. Since the the thermal analysis is a function of the material properties and they are in turn a function of the metal temperature, a coupling scenario is created. The coupling is executed iteratively by the FEA analysis until a converged thermal solid solution is obtained. The baseline blade thermal solid solution using this process is shown by Figure 56. As expected based on previous blade analyses [95], the hottest region of the blade is at the tip trailing edge and continues down a portion of the span of the trailing edge. The coolest region, intuitively, is near the base of the blade where the coolant flow is the coolest before much of the heat exchange along the inner cooling holes takes place[6].

A stress contour solution is then determined through the stress analysis step. Input for this step includes the engine rotor speed, $X_{S_{C3}}$, the metal temperature distribution from the thermal analysis, $X_{S_T}$, and the temperature dependent mechanical material properties, $X_{S_M}$, calculated using the converged temperature solution. The resulting equivalent stress contour is shown in Figure 57. Two high stress regions are found, around the mid-chord of the low pressure surface at the blade base and at the base of the leading edge. A high stress condition at the base of the blade is expected as the entire mass of the blade, and thus the blade centrifugal force, must be supported by this region. Likewise, less and less material has to be supported as you move outward along the spand of the blade so the stress is lower in that direction. An exception is the somewhat oval-shaped low stress region located between the two

[5]The heat transfer coefficient field boundary condition is actually a function of the rotational speed of the blade thus requiring the rotor speed as an input.

[6]The coolant flow enters the base of the blade, travels along the span of the blade in an outward radial direction, and exits the blade at its tip.

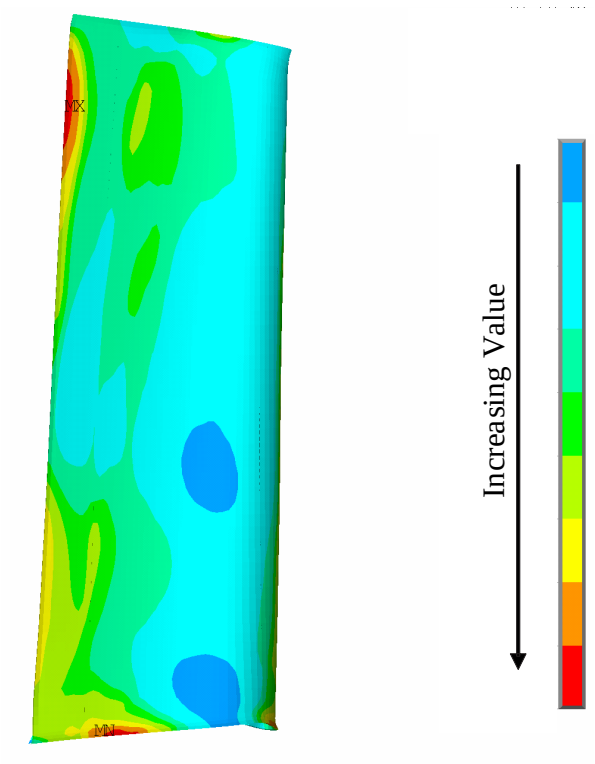


**Figure 56:** Baseline Thermal FEA Solution

high stress regions. Similar low stress regions are predicted near geometric stress concentration zones due to the material equilibrium.

Once the Mechanical analyis is complete and the stress distribution is determined, the failure analysis is then conducted. Input for the failure analysis includes the metal temperature distribution, $X_{F_T}$, the stress distribution, $X_{F_S}$, and temperature dependent material failure properties, $X_{F_M}$. The resulting overstress solution is shown in Figure 58. As the overstress is a strong function of the equivalent stress, the resulting overstress contour is very similar to the mechanical stress FEA solution where two high overstress regions are almost coincident with the high stress regions shown in Figure 57. Slight dependence to the thermal solution is expected since the ultimate stress, the denominator of the overstress ratio, is a function of temperature. This dependence, although apparently small, is shown by the moderate overstress region near the lower trailing edge tip of the blade but away from the high stress

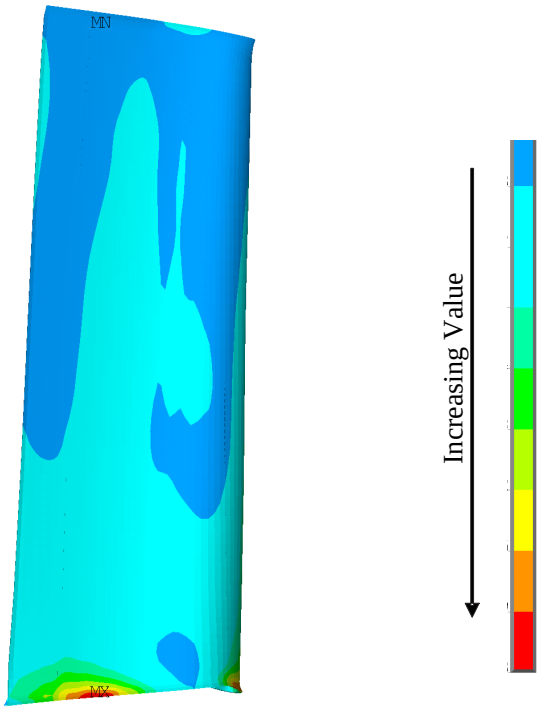


**Figure 57:** Baseline Mechanical FEA Solution (Equivalent Von Mises Stress)

regions shown in Figure 57. The fatigue life solution, shown in Figure 59, follows a similar pattern. Low fatigue life regions, represented by the dark red zones, are coincident with the high stress regions shown in Figure 57 and a moderately low fatigue life region also appears near the lower trailing edge region away from the stress hot spots. In addition, a small but moderately low fatigue life region can be seen at the trailing edge tip region of the blade as well.

The creep rupture solution requires input from all three upstream component-specific analyses. One would suspect that the limiting region of the blade due to creep rupture would be the thermal hot spot identified in Figure 56. However, several areas potentially limited by creep rupture are revealed as shown in Figure 60. Notice that limiting regions are found at both of the high stress regions as well as the high temperature region of the blade. Further, moderate to high regions span a much wider area of the blade such as the entire lower region of the blade trailing edge.

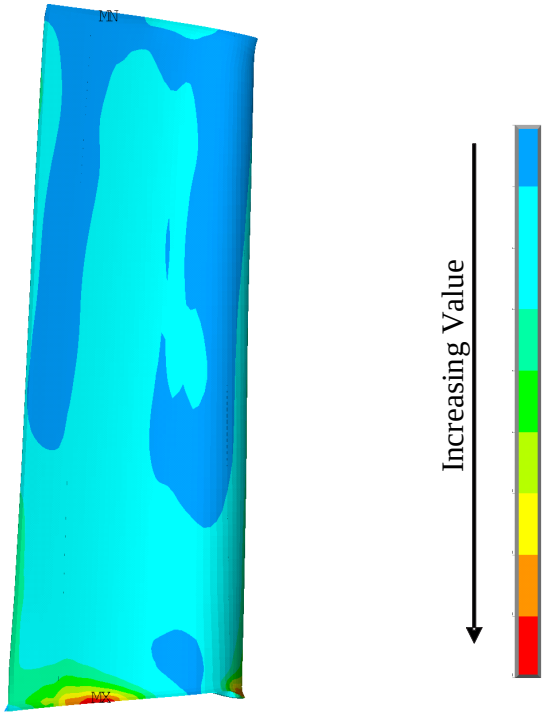


**Figure 58:** Baseline Overstress Solution

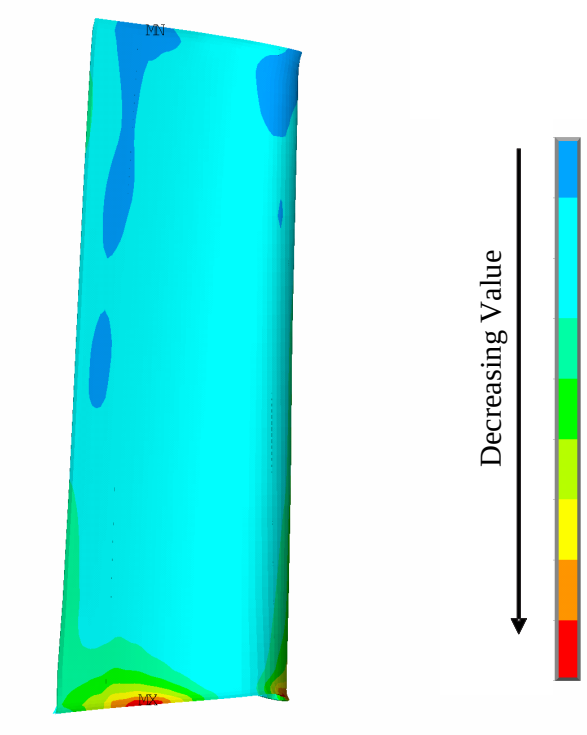


**Figure 59:** Baseline Fatigue Solution

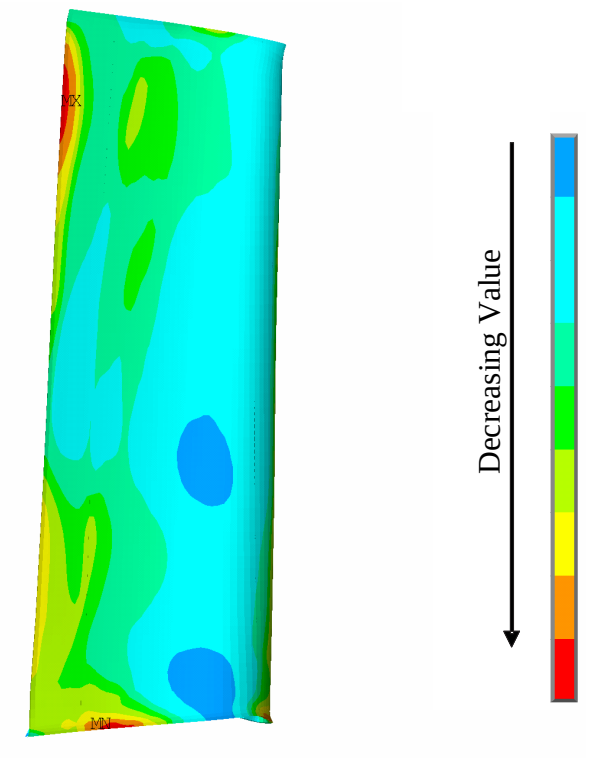


**Figure 60:** Baseline Creep Rupture Solution

The wide area covered by these limiting creep rupture regions could complicate the adequate selection of hot spots as they may migrate across these regions for different values of the upstream blade analysis input. A thorough hot spot selection process is discussed and demonstrated in section 5.5.5.

### 5.3.6 Automated BLade failure Environment (ABLE)

The baseline analysis discussed previously can be extremely tedious to perform. Several steps are involved with the process requiring a tremendous amount of dissimilar information along the way. Repeating the analysis for off-baseline cases where different values of the input parameters are to be considered is very tedious. Not withstanding, the computational expense is high as the elapsed time during the analysis is almost half an hour. To improve the efficiency of re-running the blade failure analysis as well as mitigate user error in doing so, an Automated BLade failure analysis

Environment (ABLE) has been created.

ABLE is a central analysis structure and execution engine that creates the unique input properties for each case under consideration and automatically executes the complicated analyses using this input. The rendition of ABLE used in this study was created to automate the component-specific analysis process depicted in 55. Figure 61 illustrates how the ABLE environment functions. It was constructed using a combination of PERL and ANSYS Parametric Design Language (APDL) program code. At the heart of the engine is a single PERL run script that orchestrates a set of front line PERL scripts, which interface with the pre-solved baseline ANSYS input, the user defined case input, and each of the major contributing failure analyses. For each case, the PERL scripts extract the current input values and use these to create a new ANSYS input deck by changing the respective values within the baseline failure analysis files. Then each of the analyses are executed to calculate the failure response for that case. An enture list of cases can be proved to ABLE in which it will proceed to solve for the failure response for each of them. Numerous ANSYS APDL programs, known also as ANSYS macros, were created with embedded parametric design language commands to facilitate the automation of the ANSYS thermo-mechanical analysis.

The major steps of the PERL automation are listed below:

1. Convert Coded DOE Table to Absolute Values: *convert.pl* (Appendix C.1.2.1)

2. Create case 'i' material property file: *mchange.pl* (Appendix C.1.2.2)

3. Create case 'i' external thermo-mechanical boundary condition field: *bchange.pl* (Appendix C.1.2.3)

4. Create case 'i' ANSYS Run Macros: *rchange.pl* (Appendix C.1.2.4)

5. Execute all scripts, Calculate Case 'i' failure results: *run.pl* (Appendix C.1.2.5)

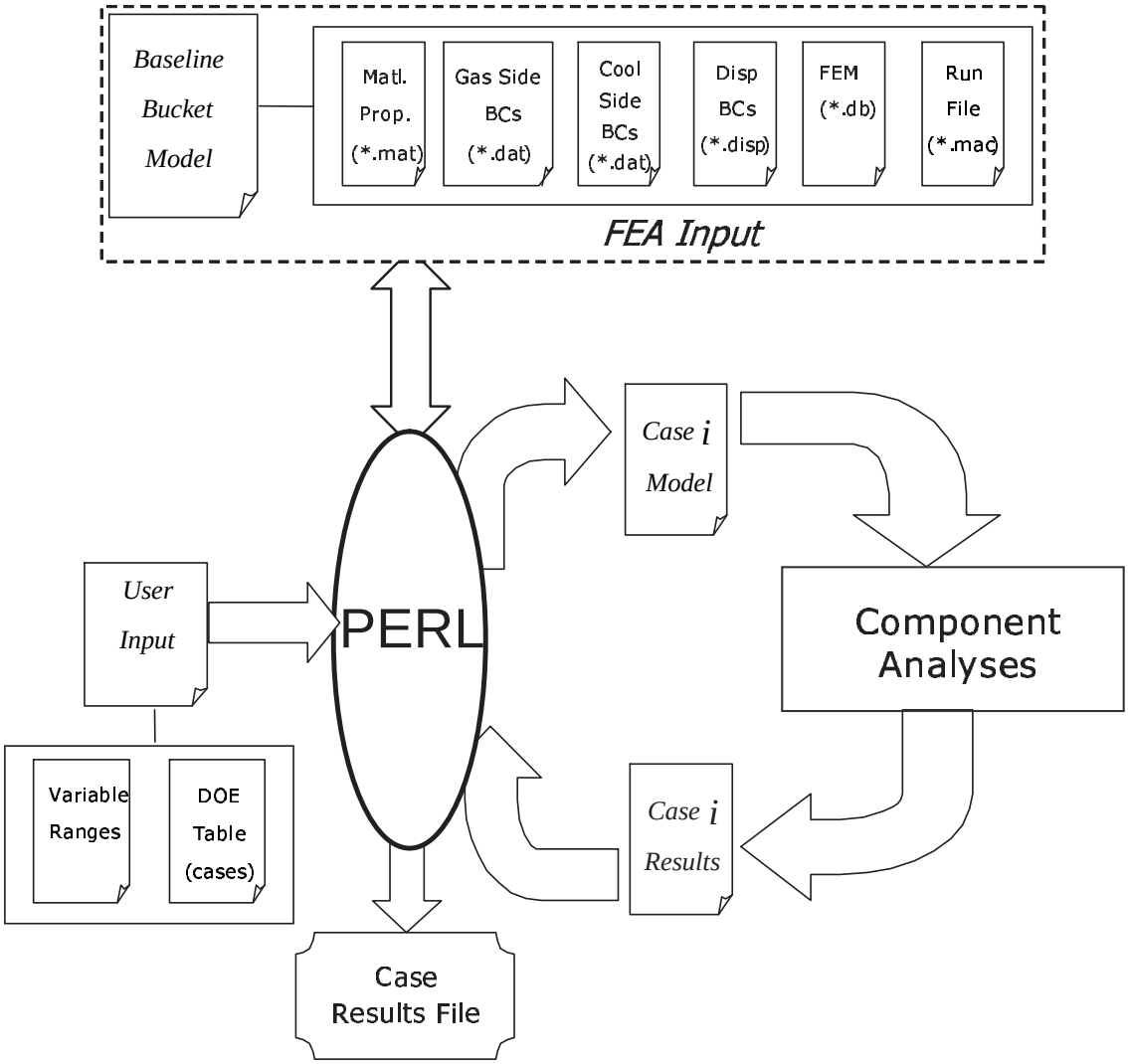


**Figure 61:** Automated BLade failure Environment (ABLE)

Material properties are created during each case by changing the room-temperature material property values to the new case value, computing the percent change in the value, and applying this percent change to the entire array of temperature dependent values of the parameter across the range of temperatures considered. Thermal boundary conditions are specified for the new case by applying two scalars, one to the external core flow boundary conditions and one to the internal coolant flow boundary conditions, computed by finding the ratio between the current cycle model core flow and coolant flow temperatures and the temperatures from the baseline cycle parameter values. For each case, these scalars are computed and used to uniformly modify the entire field of thermal boundary conditions. This approach is justified given the capability of the analysis scheme in that the cycle model output is a 1-D parameter while the blade thermal BC input is 3-dimensional. The thermal boundary condition heat transfer fields are also modified by the new rotor speed as the magnitude of heat transfer is affected by the rotational velocity of the blade. Newly computed thermal material properties are provided to the thermal analysis to find the thermal solid solution. Once the thermal solid solution is found it is then read back into ANSYS as input to the mechanical analysis. The new rotor speed is also specified, and the displacement and pressure boundary conditions applied along with the newly computed temperature-dependent mechanical material properties. The displacement constraints and pressure distribution boundary conditions are assumed constant for this study. A constant displacement contraint specification is intuitive as the interaction between the blade root and the rotor disc attachment points is not available. While stage pressure predicted by the cycle model would vary, the affect on the stress solution is extremely small compared to the rotor speed and therefore the variation in the pressure distribution is not modeled. Finally, all three failure functions are embedded directly into the mechanical run file so that the failure analysis can be run subsequent to the mechanical analysis. Material property values as a function

of temperature are determined, for each case and iteration, through a linear interpolation routine based on the material property input file. This type of modeling and simulation environment has been successfully applied to other gas turbine engine applications as well [96][94][58].

## *5.4 Local Statistical Space Assessment*

As described in chapter 4, following the synthesis of the actual component analysis environment the upstream system level analysis(es) are(is) to be queried in such a way as to confidently, yet efficiently characterize and quantify the statistical properties of the local component analysis input. This information is important to several subsequent steps such as capturing any joint randomness that may be present between the local input variables or deciding what range to consider for each local component analysis input parameter. Modeling joint randomness can be extremely important, as shown in Chapter 3, to the accuracy of the failure probability solution.

For this application there is only one system-level analysis providing input to the component-specific analyses. However, in practice, multiple upstream system analyses, other sub-system or even component analyses could provide input to the component being analyzed. Therefore, the efficient joint probability modeling methods explored in chapter 3 are applied to characterize the joint randomness of the cycle model output even though the minimal run-time of the cycle model permits an accurate large sample Monte Carlo simulation analysis. Before any of these methods are applied a baseline large sample Monte Carlo uncertainty analysis is conducted.

### 5.4.1 Baseline Monte Carlo Uncertainty Assessment

A cycle model Monte Carlo simulation using 100,000 samples was chosen and executed using the previously specified cycle model input parameter statistics given in Table 13. Statistical independence is assumed between each of the input parameters which are also assumed to be normal random variables. For each Monte Carlo sample,

**Table 15:** Cycle Model Baseline Monte Carlo Statistics.

| Parameter | Mean/Median | Deviation | Skewness | Kurtosis |
|---|---|---|---|---|
| $T_c$ | 1255.0/1253.6 | 38.8 | 0.25 | 3.19 |
| $T_g$ | 2410.7/2407.4 | 63.6 | 0.33 | 3.29 |
| RN | 9589.3/9483.7 | 1310.7 | 0.49 | 3.32 |

a sample from each random variable parameter is chosen using the inverse solution of its cumulative distribution with input from a uniformly distributed random number generated once for each parameter for each sample. The input vector for each sample is then provided to the cycle model and the required cycle responses computed repetitively until all 100,000 samples are evaluated. Given enough samples, accurate quantification of the response parameter statistics and probability space is realizable.

Statistical evaluation of the response samples is rather straightforward using most technical computing software. The statistics of the cycle model parameters is given in Table 15. These statistics suggest asymmetry, non-zero skewness, and positive kurtosis[7], $k > 3.0$, and therefore indicate that these response parameters are non-normal. Applying the Bera-Jarque hypothesis test of normality [49] quantifies this indication as the null hypothesis for normality is rejected for all three parameters.

Using the empirical CDF of the data for each response, one can begin to assess the adequacy of several candidate univariate parametric distributions (discussed in section 2.4.1). Probability plots between the empirical CDF and the lognormal, normal, Weibull, and loglogistic distributions is given in Figure 62. The Anderson-Darling test statistic is also computed and provided in this Figure for each distribution, parameter combination. A visual inspection of the graphs suggests that none of the candidate distributions fit the rotor speed data very well. Significant deviation of the lower tail exists for all of the plots and deviation of the upper tail is apparent for the loglogistic

[7]The normal distribution is symmetric and therefore produces a skewness value of 0. The kurtosis of a normal distributed random variable is equal to three where values less than three and values greater than three indicate negative and positive kurtosis, respectively.

distribution. Overall, the normal and lognormal distributions appear to fit the core flow and coolant flow temperatures better than the other distributions. Reviewing the Anderson-Darling test statistics for each parameter-distribution pair confirms the result of the probability plots. The lognormal distribution has the lowest A-D values for all three of the response parameters and thus is the most appropriate distribution, of the four considered here, for the cycle responses. Realizing that if the data follows a lognormal distribution, then a logarithmic transformation of the parameters should follow a normal distribution and thus enable the use of the Bera-Jarque hypothesis test for normality. Applying the B-J test to the log of each response parameter still, however, results in the null hypothesis of normality being rejected. The result of this finding is that although the lognormal and even the normal distributions are reasonable distributions for the response parameters, statistically they cannot be proven as the actual distribution and therefore it is likely that these three parameters follow a nonstandard distribution.

Further statistical analysis of the response parameters indicates the existence of strong positive correlation between all of the parameters. The correlation coefficient matrix for the responses is calculated to be $R = \begin{bmatrix} 1.00 & 0.85 & 0.76 \\ 0.85 & 1.00 & 0.71 \\ 0.76 & 0.71 & 1.00 \end{bmatrix}$. Thus, not only are the cycle response parameters non-normal, they are also jointly distributed.

According to the results of the baseline uncertainty assessment, assigning standard univariate parametric distributions to each of the response parameters and assuming independence would not be statistically justifiable. Alternatively, one could link the cycle model directly to the component specific analyses and run the probabilistic study in a fully integrated way; but the cycle model is a fast running representation of what would in practice actually be a time-consuming upstream analysis environment and thus a more efficient process is desirable. Several of the methods from chapter 3 are

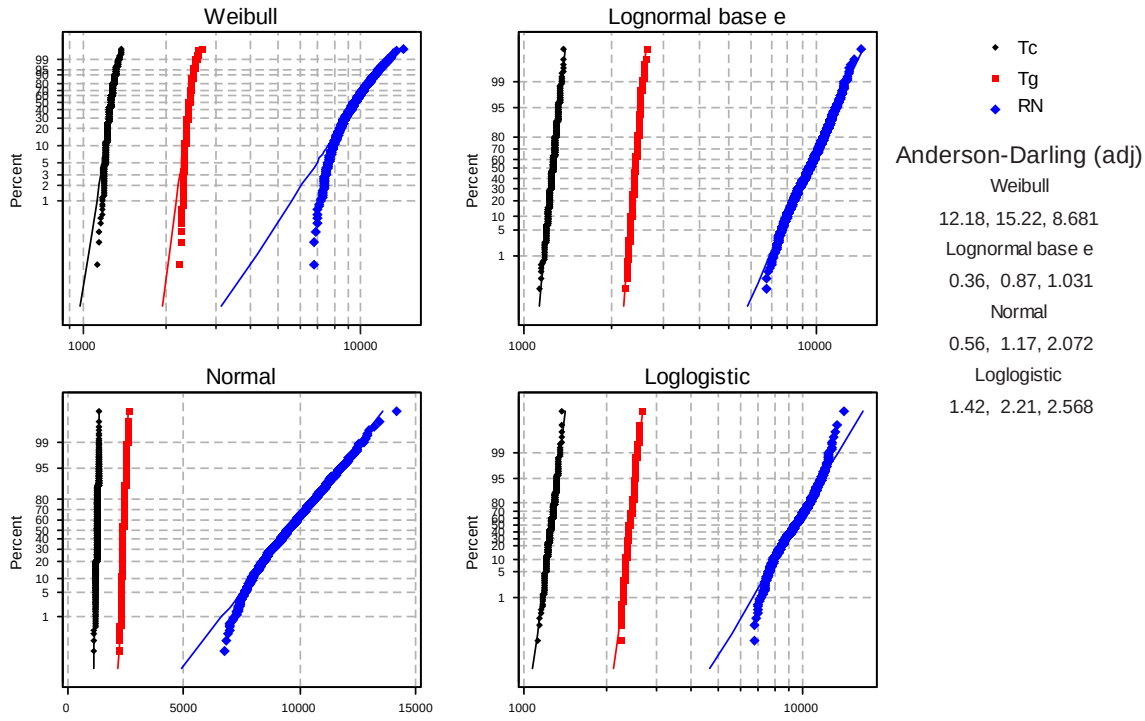


**Figure 62:** Cycle Response Parametric Distribution Identification

now considered as a means of efficiently quantifying the joint probability of the cycle parameter space. The adequacy of these models are assessed using these baseline Monte Carlo statistics.

### 5.4.2 MFOSM Model

The covariance approximation formulae derived in chapter 3 are a very efficient means of quantifying first order second moment information of a joint probability space. Applying them to the cycle model is relatively straightforward in that only $n+1$ ($n$ perturbation plus 1 mean value cases) cycle model evaluations are required to provide the necessary input to this approach. The choice of the perturbation distance is taken to be $+1\sigma$ for each of the input variables listed in Table 13. Sensitivity derivatives between each response and each of the input parameters can then be computed. Using the sensitivity derivative matrix as input to the covariance approximation formulae

**Table 16:** Engine Cycle Statistics Using Direct Covariance Approximation.

| Parameter | Mean (% Error) | Standard Deviation (% Error) |
|---|---|---|
| $T_c$ | 1248.7 (0.5) | 38.1 (1.8) |
| $T_g$ | 2401.4 (0.4) | 61.7 (3.0) |
| RN | 9474.3 (1.2) | 1299.1 (0.9) |

the covariance matrix of the response variables is found to be

$$\Sigma = \begin{bmatrix} 1450.0 & 1980.2 & 37720.7 \\ 1980.2 & 3803.3 & 56751.9 \\ 37720.7 & 56751.9 & 1687572.8 \end{bmatrix} \tag{90}$$

and the corresponding correlation coefficient matrix of $R = \begin{bmatrix} 1.00 & 0.84 & 0.76 \\ 0.84 & 1.00 & 0.71 \\ 0.76 & 0.71 & 1.00 \end{bmatrix}$ which agrees very well with the correlation coefficient matrix computed during the baseline validation analysis of the cycle model. The maximum error is 1% for the correlation between $T_c$ and $T_g$. The mean vector of the cycle responses is approximated by simply evaluating the cycle model using the mean vector of the input parameters specified in Table 13. Standard deviations are computed using the covariance matrix from equation (90) by computing the square root of the covariance matrix diagonal. Both the mean and standard deviation vectors are reported in Table 16. Not surprisingly, these approximations are reasonable compared to the baseline solution with an average error across all six statistics found to be 1.3% and a maximum error of 3.0% for the gas temperature standard deviation. However, without additional information only simple statistics and normal behavior can be estimated.

An initial limitation in this approach is that without apriori knowledge of the response distributions, non-normal response behavior can not be captured. There are two remedial measures that can be taken to overcome this limitation and specify

the appropriate response distributions: 1) Using small-sample statistical hypothesis testing, or 2) Using a second-order rather than a first-order approximation of the response space. The second approach will be considered for future work as the covariance approximation formulae would have to be re-derived for second-order information. Interestingly, the first approach can be undertaken with the already existing tools and information used within this study.

The procedure for implementing the first approach is to execute the analysis for a small number of cases using a matrix of randomly generated input variable values. Choosing the number of cases is arbitrary. Statistically the chance of finding the proper distribution is increased with an increasing number of cases yet a minimum number of cases is strongly desired for efficiency as compared to alternative methods. An alternative method for instance would be to perform an accurate statistical assessment of the cycle analysis using an applied statistical model (see section 2.5) combined with Monte Carlo simulation. The applied statistical model, to capture quadratic behavior, would be an 8 variable, 3-level design. A central composite design would require at least 84 DOE cases to achieve an acceptable resolution of IV. Subsequently, a large-sample Monte Carlo simulation, such as that of section 5.4.1, would then be used to accurately determine the response statistics and appropriate distributions. Therefore, a competitive implementation of the first approach would require much less than 84 cases. Already, $n + 1 = 9$ cases were required to approximate the mean and covariance statistics. Additional randomly generated cases are required. As a rule of thumb $2n = 16$ random cases are then executed for a total of 25 cases.[8]

The probability plots and corresponding Anderson-Darling test statistics for each response using the small-sample randomly generated cases are shown in Figure 63.

[8]This rule of thumb value has additional relevance as the second order approach would require, as a minimum, $2n$ cases.

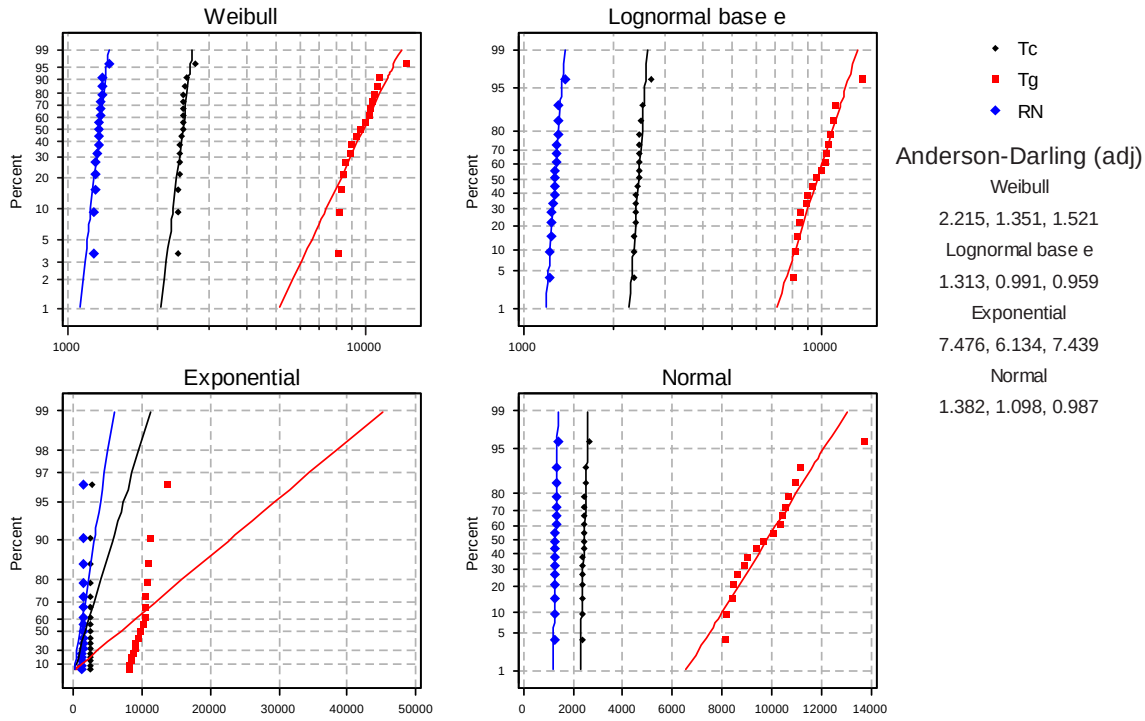


**Figure 63:** Small Sample Cycle Response Parametric Distribution Identification

Even using a limited number of randomly generated cases one can demonstrate that the lognormal distribution function is the most appropriate for the cycle analysis response variables as it had the lowest A-D test statistics for each response. Thus, even with additional randomly generated cases this approach for this problem is more efficient than that of combining applied statistical models with Monte Carlo simulation.

As a validation step, the lognormal joint distribution space as specified by this method can be validated by computing the marginal distribution statistics and comparing them to the baseline cycle model assessment. The corresponding estimated variable statistics for the improved MFOSM method are given in Table 17. The maximum error using the improved approach is now only 0.3% which will result in a significant improvement in accuracy of the input required for the probabilistic step of the framework implementation. Also, this additional step is quite interesting in that

**Table 17:** Improved Engine Cycle Statistics Using Covariance Approximation.

| Parameter | Mean (% Error) | Deviation (% Error) |
|---|---|---|
| $T_c$ | 1255.0 (0.0) | 38.7 (0.3) |
| $T_g$ | 2410.8 (0.0) | 63.5 (0.1) |
| RN | 9587.6 (0.0) | 1313.1 (0.2) |

it allows for a linear statistical approximation method, MFOSM or IPDF, to provide non-normal, even asymmetric statistical information.

### 5.4.3 IPDF Model

As an alternative to the covariance approximation approach, the cycle parameter probability space can be ascertained using the inverse transformation technique of section 3.4.2. Both approaches use the same sensitivity information that is garnered from perturbations of the analysis space; however, where the covariance approach directly approximates the mean and covariance of the space, the inverse transformation approach uses the inverse solution of the linearized approximation of the space to solve for the joint distribution of the responses.

Engine cycle statistics predicted using the inverse transformation approach are given in Table 18. Interestingly, the statistics predicted by this approach are almost equal to those predicted using the covariance approximation technique. An explanation of this finding is that the result follows well known probability theory. Remember that the inverse transformation method uses a linearized system of equations to represent the multi-response space. And, according to probability theory, a random variable that is a function of a sum of normally distributed random variables will itself also be distributed normally. In addition, the skewness and kurtosis which can be calculated using this technique were found to be approximately zero and near three, respectively. Thus, neither the covariate approximation nor the inverse transformation techniques can predict asymmetric behavior of the cycle model

**Table 18:** Engine Cycle Statistics Using the Inverse Transformation Technique.

| Parameter | Mean (% Error) | Standard Deviation (% Error) |
|---|---|---|
| $T_c$ | 1248.7 (0.5) | 38.1 (1.8) |
| $T_g$ | 2401.4 (0.4) | 61.7 (3.0) |
| RN | 9474.3 (1.2) | 1299.9 (0.8) |

probabilistic response. Nonetheless, both methods are much more efficient than the conventional Monte Carlo simulation approach. The benefit of using the inverse transformation approach would be more apparent if a complex joint probability state of the input parameters existed which could directly be accounted for. The covariance approximation technique is limited to the mean and variance information of the input parameters. Also, both the distribution identification improvement step demonstrated for the MFOSM method can also be used with the IPDF model. Although the IPDF model provides the JPDF, the distributions identified might differ due to the linearization used for the IPDF model. Then, the resulting IPDF joint probability space could be integrated and the necessary statistics calculated as input to the identified response distributions.

### 5.4.4 Conventional Approach

Several methods have been explored to characterize and quantify the output cycle model parameter space as a function of top-level system input. Such a quantification has rarely, if ever, been conducted in practice when assessing physics-based component reliability involving time-consuming physical analyses. The conventional approach would be to analyze upstream system behavior at a baseline expected condition, then assume a value for the standard deviation of each system response parameter that is required for the physical component analysis. An additional assumption for mathematical and computational convenience would be that these system response

parameters are statistically independent. The effect of the assumptions of the conventional approach is explored later during the reliability assessment of the turbine blade airfoil. Before the reliability assessment can be attempted, an assessment of the sensitivity of the blade failure responses is to be made in an effort to reduce the dimensionality of the problem.

## 5.5 Parameter Space Reduction

As with any large, complex problem such as the one considered in this study, the curse of dimensionality is a daunting factor in resisting efficient non-deterministic analyses. For instance, the analysis time required for the turbine blade failure assessment is around half an hour. If one were to consider 5, 10, and 20 potential input variables for the covariance approximation probabilistic method discussed earlier, the cumulative analysis time would be 5.5, 10.5, and 20.5 hours, respectively. To create an applied statistical model (section 2.5) of the blade failure responses for 5, 10, and 20 variables might require at least 43, 1045, and 1048617 evaluations for a three-level central composite design. However, statisticians have noted that some or even most of the potential input variables are relatively insignificant in their effect on the response parameter(s). Therefore, an initial assessment of the response sensitivity to each of the potential analysis variables is recommended where insignificant parameters are identified and set to a constant value before parametric activities are conducted. Examples of parametric activities include creating an applied statistical model or conducting a probabilistic analysis around the analysis space, both of which are conducted for the blade reliability assessment in this study.

For this study, 12 blade component input[9] and 9 response parameters have been identified and are listed in Table 19. With 12 variables it is desirable ascertain the

[9]This value is found by adding the cycle response variables listed in Table 13 to the random input parameters listed in Table 14.

**Table 19:** Component Specific Parameters.

| Parameter | Category | Description |
|---|---|---|
| EX | Input | Elastic Modulus |
| KXX | Input | Thermal Conductivity |
| ALPX | Input | Thermal Expansion Coefficient |
| DENS | Input | Density |
| CN | Input | Creep Variation |
| SY | Input | Yield Strength |
| SU | Input | Ultimate Strength |
| SFP | Input | Fatigue Coefficient |
| B | Input | Fatigue Exponent |
| RN | Input | Rotor speed |
| Tc | Input | Coolant Flow Temperature |
| Tg | Input | Core Flow Temperature |
| Tc1 | Response | Region 1 Creep Temperature |
| Sc1 | Response | Region 1 Creep Stress |
| C1 | Response | Region 1 Creep Rupture Life |
| Tc3 | Response | Region 3 Creep Temperature |
| Sc3 | Response | Region 3 Creep Stress |
| C3 | Response | Region 3 Creep Rupture Life |
| Tf3 | Response | Region 3 Fatigue Temperature |
| Sf3 | Response | Region 3 Fatigue Stress |
| F3 | Response | Region 3 Fatigue Rupture Life |

significance of these parameters with respect to their effect on the failure responses. Several methods for reducing the space of consideration are now demonstrated. In either method, the result is a reduction in the parameter space which can greatly improve the probabilistic analysis step to follow.

### 5.5.1 Single-Response Sensitivity Measures

Deterministic response sensitivity derivatives of the blade failure space can easily be calculated using the gradient approximation method described in section 4.7. The method uses a central finite difference technique to approximate the partial derivatives between the response and all of the blade failure input parameters where the perturbation length is set to $+1\sigma$ for each variable. Partial derivatives were computed

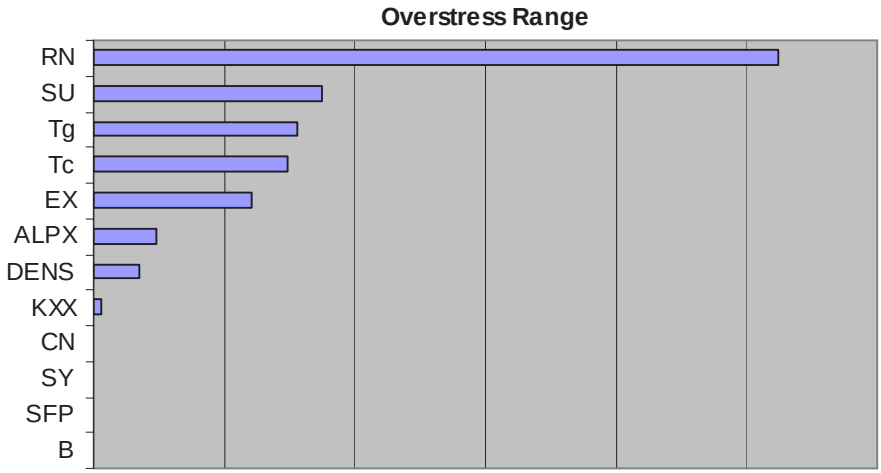


**Figure 64:** Overstress Sensitivities

for each failure mode across the span of input parameters. The sensitivity of each response to the input parameters is given in Figures 66, 65, and 64. All three failure responses evidently are extremely sensitive to the $+1\sigma$ variation of the rotor speed. Thus, the local equivalent stress is the primary driver for all three responses, more so than the local temperature. The overstress failure response (Figure 66) is also moderately sensitive to the ultimate strength (SU), the core flow (Tg) and coolant flow (Tc) temperatures, and the elastic modulus of the material (EX). The material constants, CN, SY, SFP, and B, have no effect on the overstress response as they are not used in its calculation. Also, the two fatigue constants are strong factors in the variation of the fatigue failure response. And, the temperature parameters are of the same magnitude as the rotor speed in terms of their effect on the variation of the creep rupture time response. However, the creep variation constant, CN, is only slightly important compared to the significance of the overstress and fatigue constant parameters.

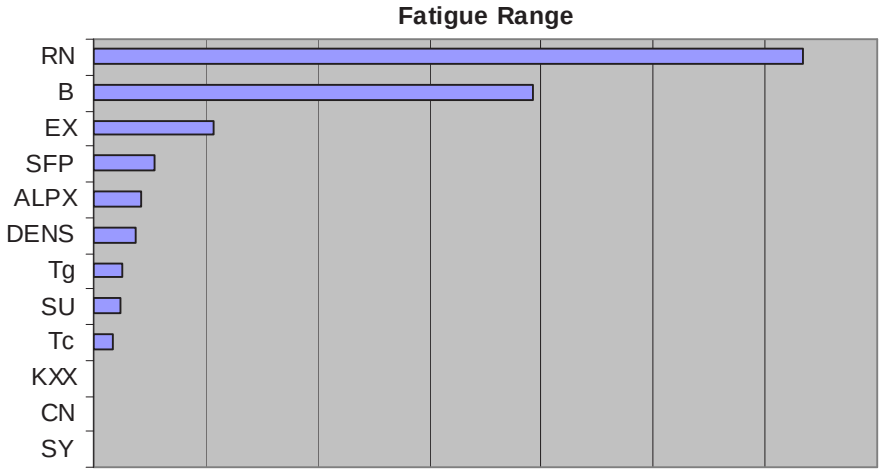


**Figure 65:** Fatigue Sensitivities

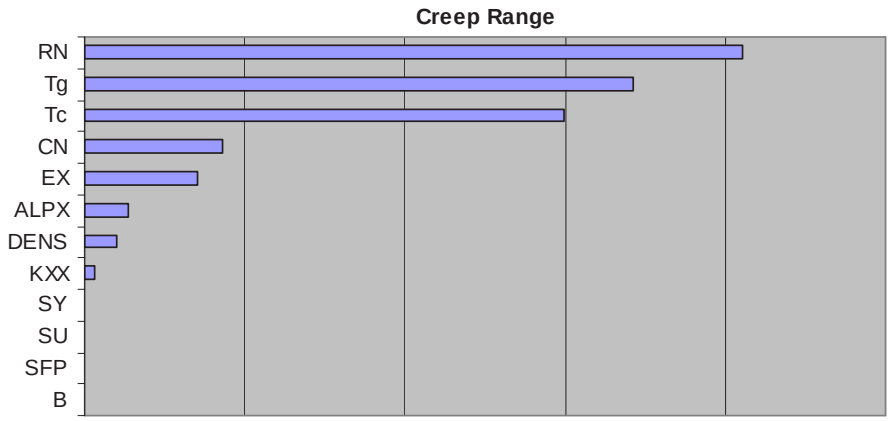


**Figure 66:** Creep Rupture Sensitivities

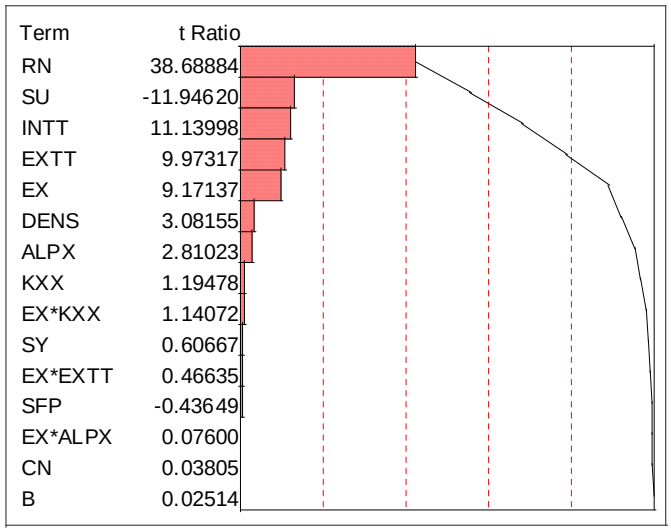


**Figure 67:** Overstress Pareto Plot

### 5.5.2 Single-Response Pareto Screening

Pareto screening is a popular applied statistical modeling approach to identifying and removing insignificant input parameters. It is analogous to the single response sensitivity measure in that the result is basically a sensitivity of each response to the vector of input parameters. However, a screening design of experiments table is created and a linear statistical model is fit to the DOE table response results. The screening DOE requires only a few cases as a 2-level fractional factorial design is commonly used. For this study, a 2-level, 17 case screening DOE was assembled and evaluated. The pareto screening results for all three responses are shown in Figures 67, 68, and 69. Overall, similar results are obtained as that of the deterministic sensitivity screening. However, neither of the methods can directly capture the statistics, particularly the input correlations, and have to be applied separately to each response.

### 5.5.3 Deterministic Spatial Reduction

Often, the limiting failure point for each failure response is identified early, perhaps prematurely, using the baseline result of the primary intermediate parameters, such as stress and temperature, before even the failure response is computed. As section

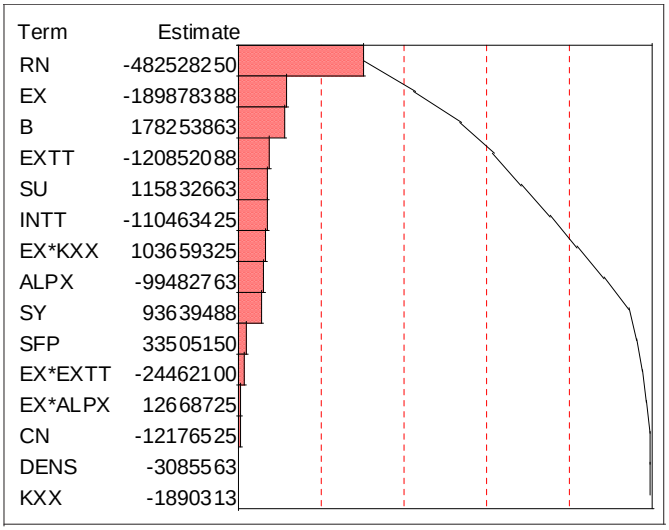


**Figure 68:** Fatigue Pareto Plot

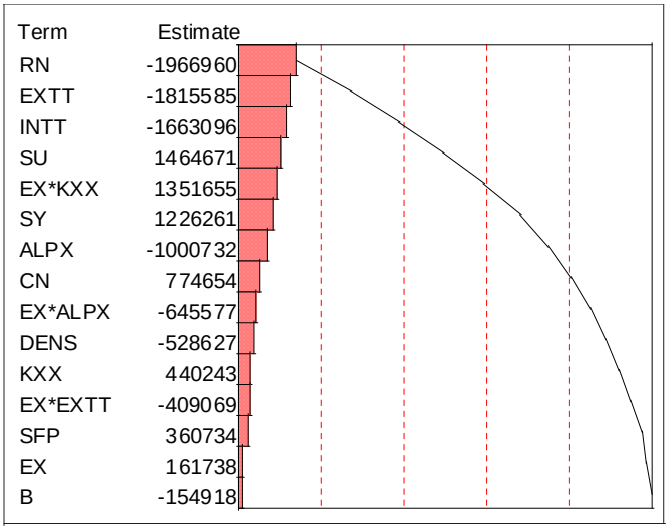


**Figure 69:** Creep Rupture Pareto Plot

5.3.5 demonstrated for the single baseline case, the actual limiting failure regions of the baseline solution can be overlooked if one chooses the worst-case values of the intermediate parameters of stress and temperature. Further what if the limiting failure point migrated or even jumped around the volume of the component as the input parameters change?

Now that the failure response space has been sampled through the sensitivity cases generated earlier, the migration of the limiting failure points for each response can be studied. For instance, the baseline creep rupture solution shown in Figure 60 points to the trailing edge tip as the limiting failure point. However, for at least one of the sensitivity cases the limiting creep rupture point jumps to the stress hot spot around the midspan of the base of the blade. Therefore, as an improvement to the process of selecting the limiting failure regions, an algorithm has been created and used within the FEA analysis to pin point the limiting failure elements over all the sensitivity cases. Further, since the sensitivity cases only represent a small sample of the actual response space, the algorithm instead of selecting a single element for each response at each case, selects all of the elements within 80% of the limiting failure response value as illustrated by Figure 70. The resulting failure region finite elements using this algorithm are shown in Figure 71. Interestingly, the algorithm selects four potential limiting regions for the creep rupture failure response. The 3rd and 4th creep rupture regions are almost coincident with the limiting regions of fatigue and overstress. Tracking all of these regions can be considered tedious, however, additional extracting additional results from the finite element solution are relatively simple to implement and can prevent the analyst from missing a limiting failure value and area during a subsequent parametric study. An additional safeguard would be to track the minimum value of each response and ensure that one of the identified failure regions from subsequent studies contained this limiting value.

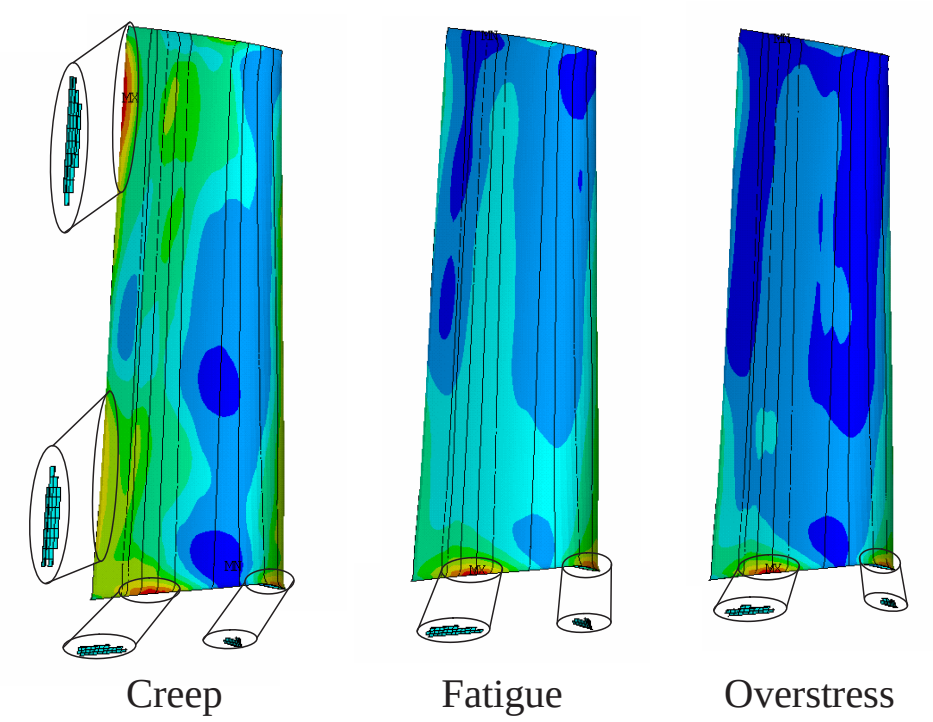


**Figure 70:** Identification of FEA Failure Hot Spots

### 5.5.4 Non-Deterministic Spatial Reduction

One of the interesting benefits of applying the covariance approximation technique within the demonstrated framework is that not only is the second-moment statistical information available now for singular parameters as reported earlier but it is also useful for calculating statistical information of field properties. Since the FEA sensitivity solutions are available and can be recorded for each element as a function of the same input parameter variations, then the covariance approximation technique can also be applied to the FEA response at each element. Therefore, statistics such as the standard deviation of the basic results such as temperature and stress or even the failure responses can now be approximated in a spatial sense. This information is invaluable in a physics-based component reliability analysis. For instance, regions of the component, or blade in this implementation, that have high values of standard

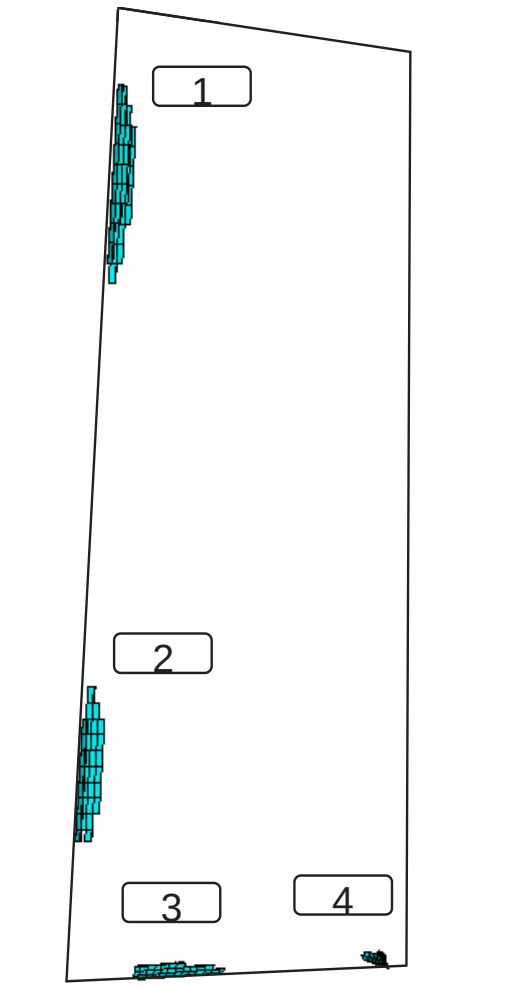


**Figure 71:** Failure Hot Spot FEM Zones

deviation can be identified and measures taken to reduce this variation both for improving the analysis prediction but also as a quality control measure to reducing the spatial uncertainty of the part.

The covariance structure of the failure responses as a function of the blade input parameter covariance structure has been calculated for each element of the turbine blade. The standard deviation of the blade temperature and stress is shown in Figures 72 and 73, respectively. Spatial behavior of the temperature variance is not intuitive as the high variance region do not overlap the high mean region shown in Figure 56. An explanation would be that the input heat transfer field is a function of the rotational speed of the bucket. Since this quantity increases outwardly along the radial direction of the blade, the variance of the heat transfer field will also behave similarly which might explain the temperature solution obtained. The variance contour of stress (Figure 73) is more intuitive as the high variance regions are aligned with the high mean stress regions shown in Figure 57. Because of the concentrated high stress value regions, remedial measures can be focused on reducing the variance in this small area.

The variance contour solution for the creep rupture and fatigue life failure responses is shown in Figures 74 and 75, respectively. A very interesting result is obtained for the creep rupture variance solution. The high variance regions are much larger than the high mean value regions of Figure 60 and, in fact, relatively more severe along the lower trailing edge of the blade. The stress variance contour solution follows the mean value solution closely.

Interestingly, since the mean value and covariance information at each element for each failure mode is now available, early spatial probabilistic information can be computed. Initial inspection of the statistical overstress failure condition suggests that the limiting condition of the stress being greater than the strength of the material is highly unlikely and can be neglected. We now focus on the creep rupture

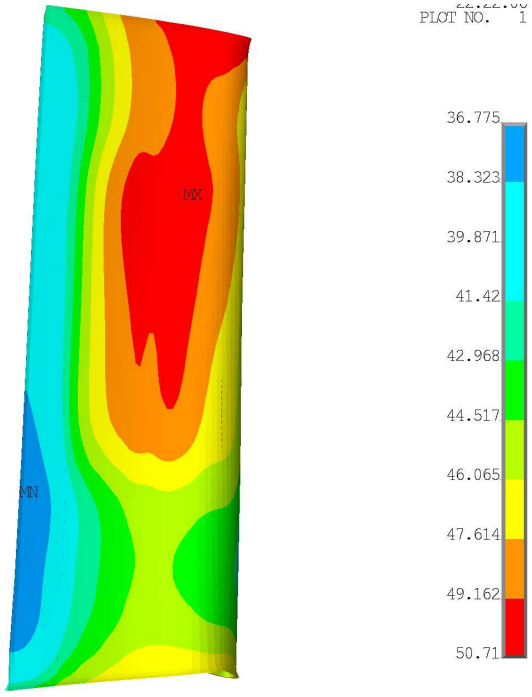


**Figure 72:** Temperature Variance Contour Plot

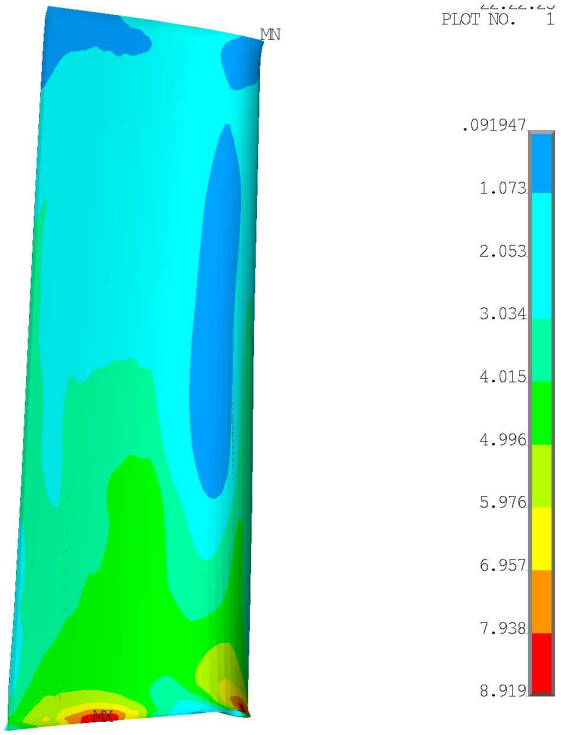


**Figure 73:** Equivalent Stress Variance Contour Plot

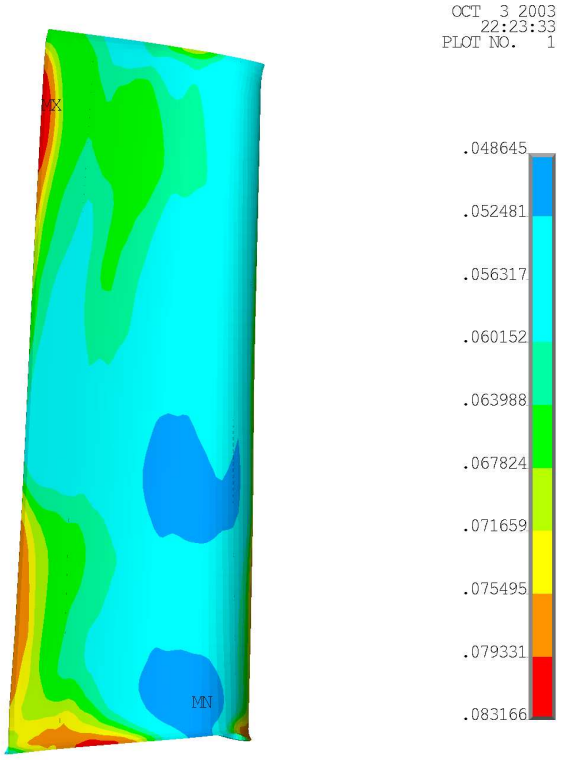


**Figure 74:** Creep Rupture Variance Contour Plot

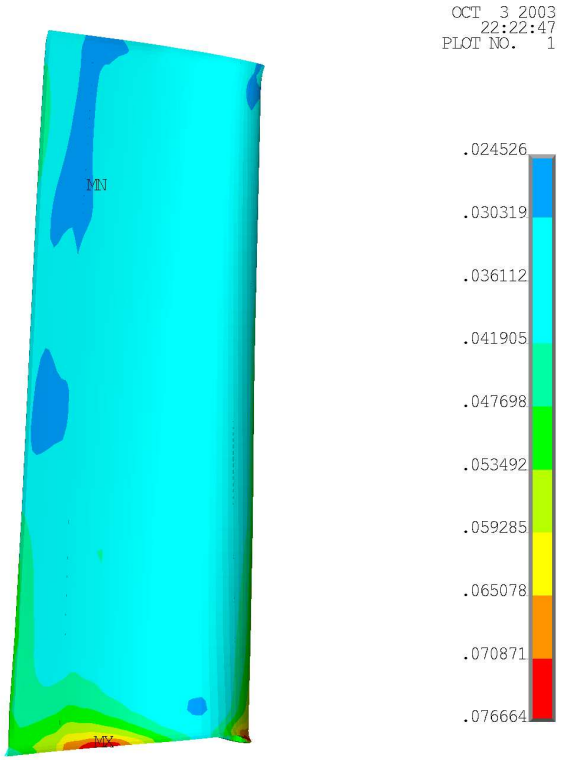


**Figure 75:** Fatigue Life Variance Contour Plot

and fatigue failure conditions. By specifying a representative limit state for both fatigue and creep and using a joint, bivariate normal distribution integration solution, the failure probability at each element can be computed. This solution is shown in Figure 76. Interestingly, the relatively high failure probability regions are highly concentrated in the two primary hot spot regions, the trailing edge tip and the base region along the low pressure surface of the blade. Plotting the failure probability contour solution is a highly useful parameter to complement the preceding deterministic hot spot identification process. *An improved hot spot selection algorithm is now available using probabilistic information in addition to the deterministic sensitivity results.* Although the non-deterministic results closely reflect the behavior of the deterministic solutions, this behavior is attributed to the rather simple spatial statistical input properties of the blade boundary conditions. The statistical variation of the field boundary conditions and material properties are assumed homogeneous for this study as non-homogeneous behavior is a rather involved behavior to model. In reality, complex non-homogenous behavior of the flow field and material field statistical properties of turbine components can exist according to Ghiocel [40] and others. Thus, future studies might show significant deviation between the deterministic and non-deterministic responses.

### 5.5.5 Multi-Response Screening Method

A new approach to the reduction of a parameter space would be to do so considering multiple responses simultaneously. This is done so within the current study by applying the approximate canonical correlation analysis (ACCA) described in section 4.7.3. The advantage of considering multiple responses simultaneously is that more information appropriate to the complexity of the problem is utilized and can thus alleviate the erroneous removal of otherwise significant parameters. In addition, the contributing analyses and even analyses input parameter subsets can also be rank

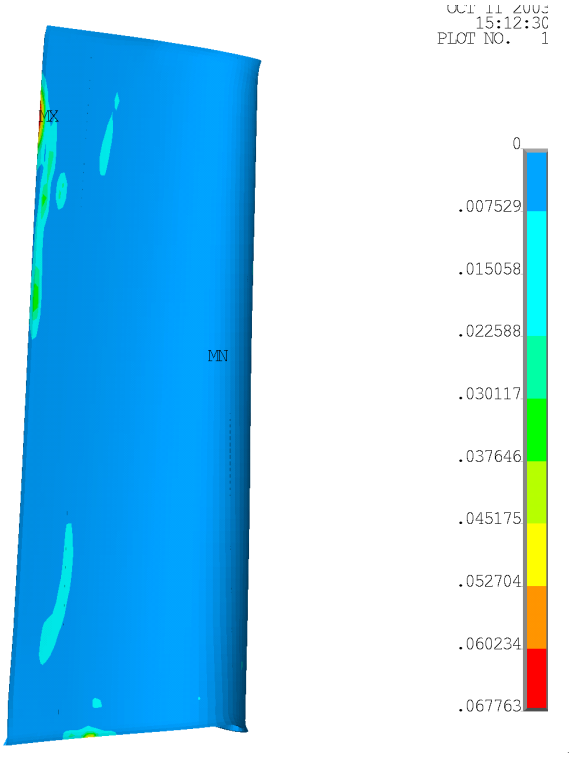


**Figure 76:** Approximate Failure Probability Response Contour

ordered and compared using this approach. These calculations and comparisons are now conducted.

The first step in ACCA is to construct the response-input composite correlation coefficient matrix. That is, a partitioned matrix is to be created by combining the correlation coefficient matrix of the failure responses, the correlation coefficient matrix of the input parameters, and the correlation coefficients between the responses and inputs, all of which would take the following compact form

$$\left[\begin{array}{c|c} R(\mathbf{y},\mathbf{y})_{9\times 9} & R(\mathbf{y},\mathbf{x})_{9\times 12} \\ \hline R(\mathbf{x},\mathbf{y})_{12\times 9} & R(\mathbf{x},\mathbf{x})_{12\times 12} \end{array}\right]_{21\times 21}$$

The elements of this matrix were computed using the sensitivity derivatives of section 5.5.1 as input to the efficient covariance approximation formulae of section 3.4.1. In this application of the covariance approximation technique, the covariance approximation results of the engine cycle parameters, $T_c$,$T_g$,and $RN$, are now used as input along

with the covariance input of the component material properties. Table 20 gives the composite correlation coefficient matrix values with the response-response correlation calculations in the upper left corner, the response-input correlation approximations in the upper right and lower left corner, and the input-input parameter correlations in the lower right corner. Strong negative correlation exists between the creep failure responses and the temperature and stress values of regions 1 and 3. This relationship is also found between the fatigue failure response of region 3 and the temperature and stress values of that region. Not surprisingly, almost perfect correlation is predicted between the temperatures of each region as well as between the stress responses of each region. Correlations between the individual response parameters and individual input parameters are also reported in the upper *OR* lower right regions of Table 20. In a sense, these correlations are analogous to the deterministic single response sensitivity calculations computed in section 5.5.1. Actually, the rank order of significance of each input parameter for each of the three failure responses, row-columns 3, 6, and 9, has resulted in very similar rank orders as compared to that of the deterministic sensitivity study. Actually, this result is expected as both the range of the responses and the response-input correlations are directly proportional when the input is uncorrelated. Although there is strong correlation between the two fatigue input parameters and the cycle parameters, the majority of the off-diagonal terms are zero. Should a majority of the off-diagonal terms be non-zero then it is possible that the rank order of input parameters using the response-input correlation calculations could differ greatly from that of the deterministic sensitivity method. Finally, the response-input correlations are a more effective metric of input parameter screening in that they reflect a combination of deterministic sensitivity as well as the correlation structure of the system. But, an even more effective measure is now computed using ACCA.

**Table 20:** Blade Parameter Composite Correlation Matrix.

| | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.0 | 0.7 | −1.0 | 1.0 | 0.7 | −0.9 | 1.0 | 0.7 | −0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | −0.6 | −0.9 | −1.0 |
| 0.7 | 1.0 | −0.8 | 0.8 | 1.0 | −0.9 | 0.8 | 1.0 | −0.8 | −0.2 | 0.0 | −0.1 | −0.1 | 0.0 | 0.0 | 0.0 | 0.0 | −1.0 | −0.8 | −0.7 |
| −1.0 | −0.8 | 1.0 | −1.0 | −0.8 | 1.0 | −1.0 | −0.8 | 0.6 | 0.1 | 0.0 | 0.0 | 0.0 | −0.2 | 0.0 | 0.0 | 0.0 | 0.8 | 0.9 | 0.9 |
| 1.0 | 0.8 | −1.0 | 1.0 | 0.8 | −1.0 | 1.0 | 0.8 | −0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | −0.8 | −1.0 | −1.0 |
| 0.7 | 1.0 | −0.8 | 0.8 | 1.0 | −0.9 | 0.8 | 1.0 | −0.8 | −0.2 | 0.0 | −0.1 | −0.1 | 0.0 | 0.0 | 0.0 | 0.0 | −1.0 | −0.8 | −0.8 |
| −0.9 | −0.9 | 1.0 | −1.0 | −0.9 | 1.0 | −1.0 | −0.9 | 0.7 | 0.1 | 0.0 | 0.0 | 0.0 | −0.1 | 0.0 | 0.0 | 0.0 | 0.9 | 0.9 | 0.9 |
| 1.0 | 0.8 | −1.0 | 1.0 | 0.8 | −1.0 | 1.0 | 0.8 | −0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | −0.8 | −1.0 | −1.0 |
| 0.7 | 1.0 | −0.8 | 0.8 | 1.0 | −0.9 | 0.8 | 1.0 | −0.8 | −0.2 | 0.0 | −0.1 | −0.1 | 0.0 | 0.0 | 0.0 | 0.0 | −1.0 | −0.8 | −0.8 |
| −0.5 | −0.8 | 0.6 | −0.6 | −0.8 | 0.7 | −0.6 | −0.8 | 1.0 | 0.1 | 0.0 | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | −0.5 | −0.6 | 0.8 | 0.6 | 0.5 |
| 0.0 | −0.2 | 0.1 | 0.0 | −0.2 | 0.1 | 0.0 | −0.2 | 0.1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | −0.1 | 0.0 | 0.0 | −0.1 | 0.0 | 0.0 | −0.1 | 0.1 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | −0.1 | 0.0 | 0.0 | −0.1 | 0.0 | 0.0 | −0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | −0.2 | 0.0 | 0.0 | −0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | −0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.8 | 0.0 | 0.0 |
| 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | −0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.0 | 0.0 | 0.0 | 0.0 |
| −0.6 | −1.0 | 0.8 | −0.8 | −1.0 | 0.9 | −0.8 | −1.0 | 0.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.8 | 0.7 |
| −0.9 | −0.8 | 0.9 | −1.0 | −0.8 | 0.9 | −1.0 | −0.8 | 0.6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 1.0 | 0.8 |
| −1.0 | −0.7 | 0.9 | −1.0 | −0.8 | 0.9 | −1.0 | −0.8 | 0.5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.7 | 0.8 | 1.0 |

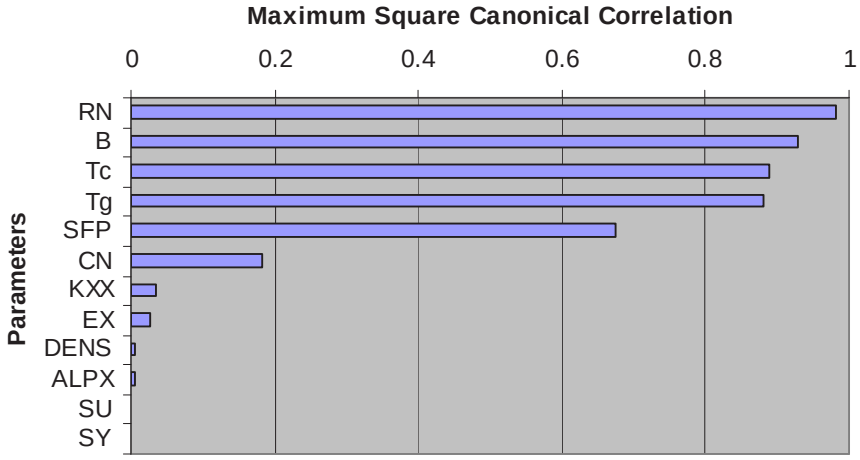


**Figure 77:** Canonical Ranking of Input Parameters

An improved screening metric is now available by using the composite correlation matrix results. Where the previous deterministic and response-input correlation screening results earlier only reflect single response rank ordering, the CCA analysis can provide the rank order in terms of how each input parameter contributes to the variation of the entire vector of response parameters. Here the response vector considered isn't all nine of the responses listed in Table 19 but rather the three failure responses, creep rupture at regions 1 and 3, and the fatigue life at region 3. The maximum square canonical correlation between the failure response vector and each individual input parameter were computed and are given in Figure 77. These canonical correlation results represent a much improved means of comparing and visualizing failure response sensitivities in a compact way and in a manner reflecting the complex deterministic and correlative information of the system. The result is that the rotor speed, RN, fatigue exponenent, B, core flow temperature, Tc, coolant flow temperature, Tg, fatigue coefficient, SFP, and the creep variation, CN, parameters are the most important contributors to the vector of failure responses of the turbine blade. An interesting difference between the CCA ranking and the deterministic sensitivity ranking is that the fatigue coefficient, SFP, moved up in order of importance for the CCA results. This is a significant result because this difference is attributed to the CCA results reflecting the increased variation of the failure response vector due to the strong correlation between the fatigue exponent, B, and the fatigue coefficient, SFP, as the input correlation between these two parameters is 0.81. Thus, the deterministic sensitivity measure cannot account for this complex statistical behavior like the CCA method can. The final result of this new multi-response method is that the blade input parameters DENS, ALPX, SU, and SY can be neglected without appreciably affecting the statistical model produced.

Additionally, how would one determine the importance of each contributing analysis? Or, within an analysis how can you rank the importance of the various sources

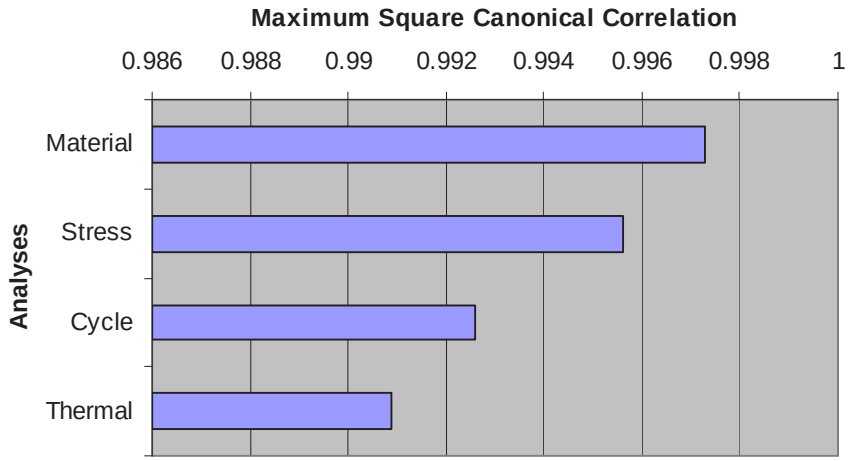


**Figure 78:** Canonical Ranking of Contributing Analyses

of input. ACCA can also provide this information. All that is required is to re-specify the vector pairs and the maximum squared canonical correlation can be computed using the same composite correlation matrix already produced. To rank order the contributing analyses, the vector of input to the failure analysis from each contributing analysis is specified and the maximum squared canonical correlation computed for each and given in Figure 78. For instance, the vector of failure response input from the cycle analysis is the rotor speed, RN, the core flow temperature, Tg, and the coolant flow temperature[10,11], Tc.This analysis produces an uninteresting result in that all of the values are greater than 0.99 suggesting that each of the analyses are extremely importance to the deterministic prediction and non-deterministic behavior of the failure responses. However, an order of importance can still be made resulting in the material analysis being the most importance analysis for the turbine blade reliability assessment.

[10]Although the actual failure response formulae are not direct functions of these parameters, the CCA method still accounts for their importance through the execution of the sensitivity analysis step within the covariance approximation technique which provides the composite correlation matrix.

[11]The flexibility of the CCA method is apparent in the case of the thermal and stress analysis as the temperature and stress solutions are actually responses. Within CCA a vector can be any combination of response and input parameters.

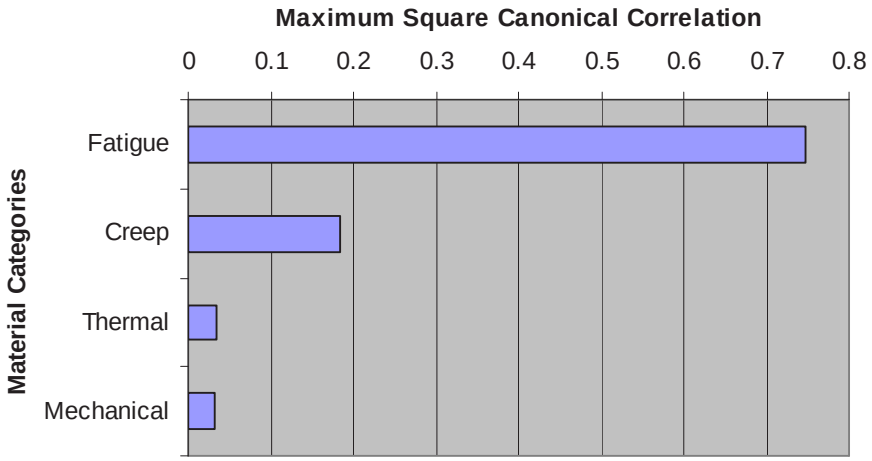


**Figure 79:** Canonical Ranking of Material Categories

Considering that the material analysis was found to be most significant, although not by much margin, then further assessments of this analysis can be made with ACCA. Actually, there are several categories of input variables to the material analysis. For instance, the variables elastic modulus, EX, and density, DENS, are basic mechanical variables. The variables fatigue coefficient, SFP, and fatigue exponent, B, are material fatigue constants and the creep variance parameter, CN, is a creep property. The thermal conductivity, KX, and thermal expansion coefficient, ALPX, are thermal material properties. Thus, a multi-response rank order can be assessed for the individual material property categories and is given in Figure 79. This rank ordering produces a more interesting result than that at the analysis level. It appears that compared to the other categories, the fatigue properties are by far the most important. Therefore, this information would be highly useful for determining where to focus performance and quality control efforts. Based on this result, one would spend resources to improve the fatigue properties and reduce their variation. This would be the most efficient way to improve the failure response characteristics within the materials arena.

## 5.6 Probabilistic Failure Assessment

The preceding sections have allowed for a sound preliminary understanding of the behavior of the blade failure characteristics. But, the primary purpose of the framework is to assess the failure probability of the component using an improved characterization of the complex statistical space of the intermediate system-component variables. This is essentially a decomposition approach where system variation is efficiently quantified using the novel methods of chapter 3 and applied to a subsequent component-level probabilistic analysis to assess the component reliability. A baseline truth solution is ascertained using large-sample Monte Carlo simulation with the linked cycle-component analysis structure matrix depicted in Figure 55. This will be labelled as the fully integrated approach. The same Monte Carlo environment without the system analysis link can be applied again at the component analysis level to find the probability solution using the conventional approach described in section 5.4.4. The probability solutions using all three of these methods are compared.

### 5.6.1 Baseline Fully Integrated Solution

The baseline Monte Carlo solution used to validate the subsequent probabilistic approach is created simply by performing a large sample Monte Carlo simulation with the full cycle-component analysis environment depicted in Figure 55, with one difference. Due to the prohibitively large dimensionality created by the FEA solution and the number of input variables, an accurate, yet fast surrogate model of the FEA solution is utilized in place of the time-consuming analysis. The surrogate or substitute model is created using the familiar applied statistical modeling technique reviewed in section 2.5.

#### *5.6.1.1 Applied Statistical Model*

The first step in creating an applied statistical model is to assume the general form of the response approximation. For instance, it is assumed here that the FEA responses

are at most quadratic and thus a second-order, eight variable polynomial function is selected (reference equation 22). Eight variables are considered rather than the full twelve to reduce the dimensionality of the problem. Sensitivity derivatives and pareto screening information generated in section 5.5 can be used to select the most important variables for the assumed polynomial function. Results calculated by the execution of a designed test matrix (DOE) are used to solve for the coefficients of the polynomial function via least squares regression. The DOE type chosen is a three-level, resolution V central composite design which for one center point will require 81 evaluations of the long-running FEA solution. The result of the model fitting process will be an accurate metamodel of the creep rupture response at regions 1 and 3, and the fatigue life response at region 3. Other potential hot spot regions and failure modes are neglected based on the results of the thorough screening study conducted earlier.

As shown in Figures 64 through 69, the sensitivity and pareto screening results suggest that two different input variable subsets should be considered for the creep rupture and fatigue life failure responses. For creep rupture, the variables chosen are EX, KXX, ALPX, SY, SU, RN, INTT, and EXTT where for fatigue the variables chosen are EX, KXX, ALPX, SY, SFP, B, RN, and EXTT. Therefore, two separate DOE tables are created, one for each of the two variable subsets.

By evaluating the failure responses for each case within the two DOE test matrices, the necessary response data is obtained for creating the polynomial functions representing the time-consuming FEA solution. Initial failure response results generated for both DOE tables were so highly non-linear that the fitted polynomial approximations, although showing a good appoximation fitness for the actual points, predicted validation errors of more than 10%. Performing several remedial measures such as dependent and independent variable transformations and re-fitting the polynomial functions proved to be unsuccessful for creating an accurate model. Inspecting the

functional form of both the creep rupture (equation 87) and fatigue life (equations 88 and 86) equations one can quickly see why they would be difficult to model using only a second-order polynomial. Both of the failure equations are a function of the FEA stress and temperature solution at the two regions considered. Therefore a polynomial approximation of the stress and temperature for the regions was created and used as an input to the failure functions?

The model fitness and accuracy of the resulting polynomial approximations of temperature and stress for regions 1 and 3 proved to be exceptional. A perfect coefficient of determination of one was obtained for all of the polynomials and validation errors using 40 extraneous, randomly generated cases resulted in a maximum error of 0.02%.

Prediction curves for the temperature, stress, and creep rupture responses for regions 1 and 3 are shown in Figure 80. Using the hairline curves between each response and each input variable, one can assess the importance of the variable by the steepness of the curve as well as the degree of linearity. For the most part, the relationships between the responses and inputs exhibit only slight to moderate curvature. The steepest curves are those between the stresses and the rotor speed, which is due to the state of stress being extremely dependent on the rotational speed of the blade. Likewise, increasing gas, Tg, and coolant temperatures, Tc, increase the failure input temperature. What may not be intuitive, initially, is the inverse relationship between the failure temperature and rotor speed as shown by the negatively sloped hairlines located at the second and fourth rows of the sixth column. However, this trend is easily explained by virtue of the aerothermal boundary conditions applied in the model. Remember that the heat transfer coefficient field is modelled as a function of the rotor speed where an increase in rotor speed reduces the heat transfer coefficient field magnitude thus reducing the amount of heat exchange between the hot gas flow field and the blade. Generally the same overall behavior is shown for the temperature,

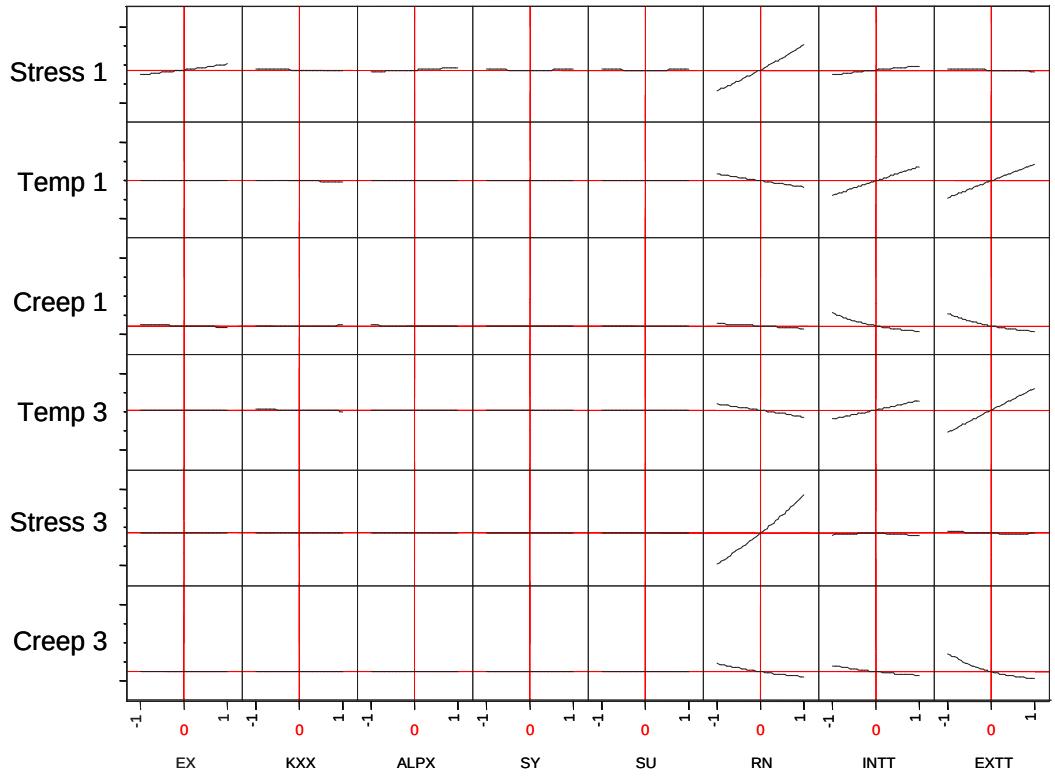


**Figure 80:** Region 1 and 3 Creep Rupture Response Surface Equations

stress, and fatigue failure responses for region 3 as shown by Figure 81.

#### *5.6.1.2 Large Sample Monte Carlo Assessment Using EDF*

Now that an accurate metamodel of the FEA thermal/stress solution has been created, the entire cycle-component analysis structure matrix can be executed effortlessly over many cases. Such is the requirement to implement the Monte Carlo simulation method where thousands of evaluations are required to generate the empirical distributions of the blade responses. This method for multiple simultaneously considered responses is the EDF method utilized for validation in chapter 3. The first step of the Monte Carlo simulation process is to select an adequate number of evaluations or simulations. Although Monte Carlo simulation is the most accurate probabilistic method [35], its accuracy in predicting event probability values decreases rapidly for lower and lower event probability values. For instance, a failure probability of 0.02%

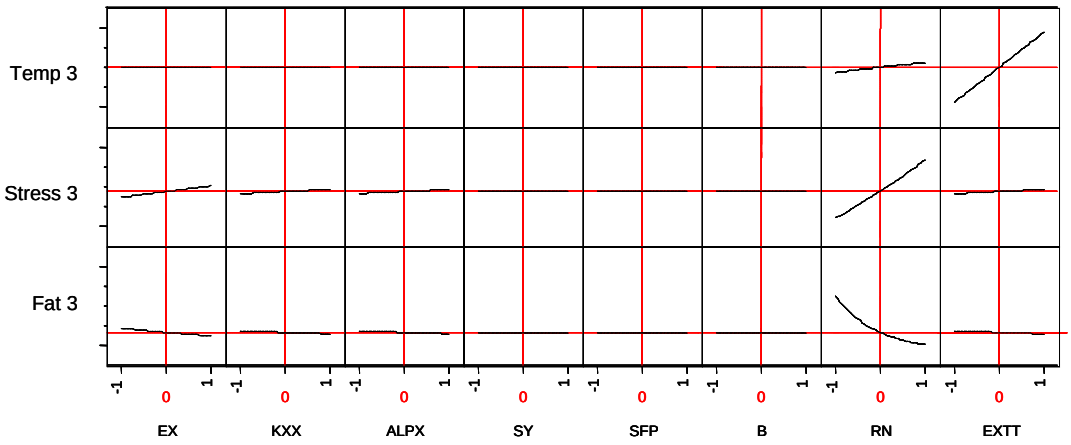


**Figure 81:** Region 3 Fatigue Failure Response Surface Equation

means that 2 in 10,000 evaluations resulted in failure. But, the problem is more difficult. Actually, the event probability calculated using Monte Carlo simulation is itself a random variable with a confidence interval and associated error. Fortunately, Shooman [79] has, using the central limit theorem, derived the distribution of this quantity and as a result has provided a formula for estimating the Monte Carlo event probability calculation error (%) given as

$$\varepsilon = \sqrt{\frac{(1 - P_f)}{N \cdot P_f} \cdot 200} \tag{91}$$

where $N$ is the number of evaluations and $P_f$ is the failure probability. Using this formula, an appropriate number of simulations can be determined to make a highly accurate calculation of the blade failure probability. Given the nature of equation (91), an initial guess of the number of simulations is made, the resulting $P_f$ value calculated, and an initial error assessment made. This process must be repeated iteratively by increasing the number of simulations until an acceptable error is achieved resulting in a converged $P_f$ solution. With an accuracy of within 0.5%, it was determined that 100,000 simulations were required to converge to the solution of $P_f = 0.0151$ at this accuracy level.

Statistical properties for all of the original top-level cycle input and component-level input variables were used as input to the Monte Carlo simulation. Failure response values were then generated over 100,000 randomly generated input vector values. This failure data set can then be used to perform the probability calculations as follows. By defining a failure event to be the case when a creep or fatigue life value is less than the time interval between aircraft engine overhaul, the appropriate and accurate failure event probabilities can be calculated using a simple counting technique. For instance, for a failure probability of 0.0151, 1,510 failure events were counted for the 100,000 cases simulated.

All of the necessary baseline failure probability calculations are given in Table 21. The first three columns report the individual failure event probabilities for each failure response. This calculation is simply a computation of the marginal failure probability of each response irrespective of any joint randomness. Creep at region 3 has the highest failure probability while Creep at region 1 is extremely small. Joint failure probabilities, the event that the blade will failure during the time interval, can be calculated by solving the following expression

$$P_{f,blade} = P\left[c3 < t \cup f3 < t\right] = P_{f,c3} + P_{f,f3} - P\left[c3 < t \cap f3 < t\right] \qquad (92)$$

which is the probability that failure occurs due to creep or fatigue at region 3, where $t$ is the specified elapsed usage in units of time, $P_{f,c3}$ is the failure probability due to creep at region 3, $P_{f,f3}$ is the failure probability due to fatigue at region 3, and $P\left[c3 < t \cap f3 < t\right]$ is the probability that both creep and fatigue failure will occur at region 3. Often, as is done in the popular 'series event' approach, the events are assumed independent which results in the simplification of the last term in equation 92 to simply the product of the two event probabilities, $P[f, c3\cap, f3] = P_{f,c3} \cdot P_{f,f3}$. The joint failure probability using the independence assumption is calculated and reported in Table 21. However, numerically equation (92) can be calculated with ease using the simple counting technique within Monte Carlo. Thus, the actual joint

**Table 21:** Baseline Blade Reliability Results.

| Individual Response $P_f$ | | | Joint $P_f$ | |
|---|---|---|---|---|
| Creep Region 1 | Creep Region 3 | Fatigue Region 3 | Independent | Dependent |
| 0.0000 | 0.0141 | 0.0079 | 0.0219 | 0.0151 |

probability solution to the third term in equation (92) can be found using an AND function, or the actual failure condition evaluated directly using an OR function within the counting technique used. The resulting joint probability solution is found to be 0.0151. The solution using independence although conservative is more than 45% from the actual joint probability solution. Further, we can see that selecting the maximum marginal distribution failure probability from the first three columns of Table 21 is not only non-conservative but also mathematically incorrect.

Additional investigation of the blade failure probability space can be made by visualizing the empirically generated data. Applying a two-dimensional statistical contour plotting routine using the EDF data one can visualize the joint probability density function (JPDF) between the two driving failure modes, creep and fatigue, at region 3. The JPDF of these variables is shown in Figure 82 where the failure responses have been normalized using an arbitrary factor. Considerable concentration of mass is observed at the lower values of both failure responses. Also, the initial cross-section shown on the left forward face of the plot illustrates what appears to be lognormal behavior for the creep failure mode. To investigate the individual behaviors of both failure responses one can visualize the empirical marginal distributions for both variables as shown in Figure 83. It appears that both of the variables exhibit lognormal behavior, which would explain the early concentration of mass found in the joint distribution plot.

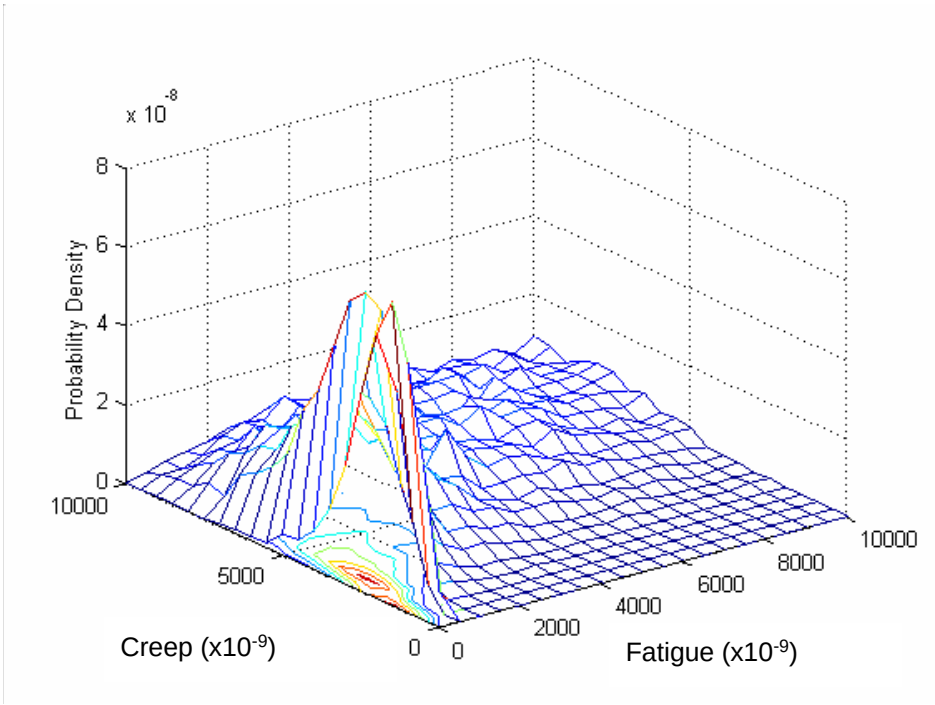


**Figure 82:** Turbine Blade Joint Failure Probability Distribution.

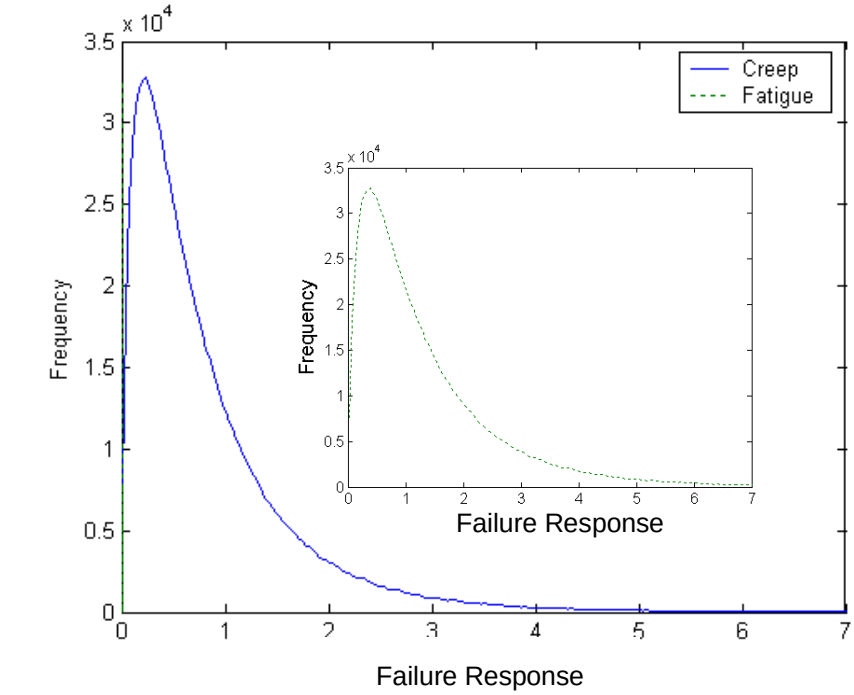


**Figure 83:** Turbine Blade Marginal Failure Probability Distributions.

### 5.6.2 Application of Sensitivity-Based Models: MFOSM and IPDF

The approach taken to produce the baseline solution to the blade failure probability was done using a fully integrated system-component environment. However, practical implementation of a fully linked environment would be prohibitive. Therefore, the Novel joint probability methods of chapter 3 are applied here and compared to the baseline solution for accuracy and efficiency.

The task is to identify all complementary analyses that are required to provide the input local to the component analysis. In this problem, the cycle model is the only upstream system analysis that provides the local input. Therefore, the statistical characterization and quantification information generate in sections 5.4.2 and 5.4.3 can now be used as input, in leu of the linked cycle model, and the component reliability assessment performed.

After specifying the the local input parameter statistics for the parameters, Tc, Tg, and RN, using the novel joint probability methods, the subsequent steps to perform the blade reliability analysis are done so using the same EDF Monte Carlo approach used in section 5.6.1. Thus, the accuracy and efficiency of decomposing the problem to system and component elements using the novel joint probability methods can be assessed. First, the mean vector and covariance matrix with the joint normal assumption is considered and the resulting blade reliability values given in Table 22. The joint failure probability is calculated to be 0.0072 which is 52% from the actual solution of 0.0151. However, by utilizing the improved small-sample statistical-based distribution identification step the results, shown in Table 23, using this joint lognormal space greatly improve the blade reliability calculation. The joint failure probability using the improved MFOSM method is 0.0147 which within 3% of the actual solution. Thus, the identification of the appropriate distribution is crucial in accurately predicting the blade reliability.

**Table 22:** Original MFOSM and IPDF Results.

| Individual Response $P_f$ | | | Joint $P_f$ | |
|---|---|---|---|---|
| Creep Region 1 | Creep Region 3 | Fatigue Region 3 | Independent | Dependent |
| 0.0000 | 0.0063 | 0.0027 | 0.0090 | 0.0072 |

**Table 23:** Improved MFOSM Results.

| Individual Response $P_f$ | | | Joint $P_f$ | |
|---|---|---|---|---|
| Creep Region 1 | Creep Region 3 | Fatigue Region 3 | Independent | Dependent |
| 0.0000 | 0.0146 | 0.0072 | 0.0216 | 0.0147 |

### 5.6.3 Conventional Approach

In practice, one would assume that the variables local to the component reliability analysis were independent and normally distributed. This conventional approach can easily be implemented using the framework developed herein. First, the off-diagonal elements of the local parameter covariance matrix are set to zero and the original normal distribution assumption applied. The results of this approach are reported in Table 24 and show that this approach can introduce considerable error in the failure probability assessment as the joint probability solution is 79% from the actual solution. An error of 79% can for this application be considered the best case scenario as the conventional method would use an ad-hoc approach to specifying the variance of each parameter where the variances here were provided by the accurate MFOSM and IPDF joint probability methods. For this reason, actual predictions using the conventional approach would likely be even more inaccurate. Additionally, the relative failure probability values of each individual response do not compare well with the actual solution. This approach fails to predict the significance of the creep failure probability at region 3. Therefore a very important result is revealed here. By neglecting the joint randomness and non-normality of the local input parameters,

**Table 24:** Conventional Method Results.

| Individual Response $P_f$ | | | Joint $P_f$ | |
|---|---|---|---|---|
| Creep Region 1 | Creep Region 3 | Fatigue Region 3 | Independent | Dependent |
| 0.0000 | 0.0000 | 0.0032 | 0.0032 | 0.0032 |

the statistical significance of creep rupture at region 3 can not be revealed. Not only will failure probability predictions be erroneous, critical decisions to control only the fatigue life response at this region could be made.

### 5.6.4 Comparison of Approaches

Each of the approaches used to conduct the component reliability calculations are now compared. The joint failure probability using each approach is reported in table 25. The baseline solution required $100,000$ evaluations of the actual cycle model to properly characterize and quantify the cycle response parameters, Tc, Tg, and RN, but is deemed the most accurate solution. The conventional approach in practice only requires one evaluation of the cycle model combined with a guess of the variable variances. However, the error of this approach in predicting the blade failure probability is very large. Also, this is the only interacting analysis to the component-specific analysis set. What if additional analyses representing other subsystem elements were to be considered? Introduction of additional error over what is currently predicted is to be expected. For instance, what if we were to include the interaction between this component and other components, or the interaction between the turbine and the compressor modules. Neglecting the complexity of the joint randomness structure can greatly affect the non-deterministic analysis results.

Fortunately, two new joint probability methods are now available to capture this complex behavior. The use of the covariance approximation technique to enable the application of either of the two novel joint probability methods requires only $n + 1 = 9$ evaluations and improves the failure probability prediction by 37%. An

**Table 25:** Comparison of Probabilistic Methods.

| Method | Joint $P_f$ (Error) | Evaluations |
|---|---|---|
| Baseline | 0.0151 | 100,000 |
| Conventional[1] | 0.0032 (79%) | 1 |
| MFOSM&IPDF | 0.0072 (52%) | 9 |
| Improved MFOSM | 0.0147 (3%) | 25 |

[1] Using parameter variances calculated from MFOSM method

extra $2n = 16$ evaluations for a total of 25 provides a substantial improvement to the blade failure prediction as the solution is within 3% of the actual solution. The novel joint probability methods therefore provide a great deal of accuracy and efficiency for this problem. Further, since these methods are accurate even for low probability calculations, one can deduce that predictions made closer to the center of mass of the joint response distribution would be even more accurate. In addition, substantial information of the intermediate parameter space is now available. For instance, we now know that the cycle response parameters follow a distribution close to the trivariate joint lognormal distribution. Therefore, future evaluations can utilize this information in quantifying the local parameter statistical properties.

## *5.7 Summary of Results*

Several results have been garnered using the formulated component reliability framework applied to a second stage jet engine turbine blade. Firstly, the probability of blade failure for this component as predicted using the framework is 0.0147 which can be stated as a prediction that around 15 out of 1,000 turbine blades could fail[12] during the initial time interval to the first overhaul. Thus, an improvement to the component design or a limit to the system operation is recommended. Also, the conventional method proved to be highly inaccurate as compared to the novel joint

[12]This failure probability prediction is for a generic model under generic conditions and does not represent an actual failure probability prediction of an actual component

probability methods utilized within this framework. One noteworthy result is that the independence assumption often used in the conventional component reliability method has been shown in this study to be highly inaccurate and misleading as to the root cause of blade failure.

Additional information was deduced from the early framework implementation steps. The same very framework techniques used to reduce the blade reliability parameter space can be used to make recommendations to decrease the blade failure probability. For instance, the new approximate canonical correlation analysis (ACCA) suggests that the material analysis module is the most influential analysis in terms of the reliability of the blade. Therefore, effort should be focused on improving the material properties through processing and preparation or even selecting a superior material. According to the ACCA, the creep and fatigue properties were deemed the most important and a material with improved fatigue and creep properties should be pursued.

The highly useful covariance approximation technique using a first-order sensitivity approximation was utilized in several areas during the framework implementation in addition to the joint failure probability assessment. This technique was used to provide the necessary composite correlation matrix to enable the CCA to be conducted. Also, the covariance matrix generated using this technique was used to provide an early spatial assessment of non-deterministic quantities of the blade such as the basic and failure response parameters as well as early failure probability. Thus, a new non-deterministic spatial reduction scheme is now available to reliability analysts. In this implementation the non-deterministic hot spots were aligned, for the most part, with the deterministic hot spots. However, should random field properties be considered and modelled, an entirely different result might be found where the non-deterministic hot spots may migrate from the deterministic ones. Such a behavior can be screened early using covariance approximation.

Finally, the framework implementation was conducted in a systematic manner. In doing so, critical information both at the system and component level is organized and generated in a way that is unique to the component being analyzed. The early decomposition and synthesis steps in the framework allow the analyst to qualitatively inspect the necessary aspects of the component reliability problem in relation to the most significant safety and reliability concerns of the component.

# CHAPTER VI

# CONCLUSIONS AND RECOMMENDATIONS

Component reliability has traditionally been conducted using overly simplified statistical assumptions without assessing uncertainty introduced through component system interaction. To address this deficiency, a novel systematic component reliability framework has been formulated and successfully implemented for a jet engine turbine blade application. The results of the implementation demonstrate the need for modelling joint randomness when attempting to quantify the failure probability of components operating in complex systems. Extreme joint randomness and non-normal behavior was found for the intermediate variables local to the component analysis. The complexity of this behavior has been shown to be an attribute of the component-system interaction. Such interaction as shown within *can* be accounted for through efficient joint probability models developed during this study.

The findings of the the framework, as implemented, and the preceding validation exercises are surveyed to draw conclusions for answering the driving research questions, emphasizing the significance of the new framework, and making recommendations for future work and applications. Section 6.1 takes an introspective look at the body of this study to answer the governing research questions first posed in section 6.1. Section 6.2 summarizes the significance of the developed framework and highlights the individual contributions made during this research study. Section 6.3 provides recommendations for improving related industrial practices, identifying additional applications of the existing framework, and recommending where continued academic research of this topic should be focused.

## 6.1 Research Questions

The research questions posed at the onset of this study are now revisited and answered based on the results generated within the body of this work. Some of these questions have been plaguing the reliability discipline for some time and just now can be answered using the developed framework and associated mathematical tools. The questions and their answers are as follows:

1. What are the limitations of currently accepted methods used in industrial probabilistic assessment activities?

   - Current industrial component reliability methods statistically isolate the component when conducting reliability assessments. This isolated approach, while greatly reducing the workload of the procedure, uses unfettered assumptions of normality and independence of the local component input parameters and neglects joint randomness of the response variables. As shown within this study, neglecting the complex statistical behavior of the input *and* response parameters can introduce significant error in component reliability assessments.

2. What statistical behavior is expected for components of complex systems, such as a gas generator turbine blade?

   - Through several examples within this study it is apparent that complex statistical behavior is to be expected for components operating within a system. Such statistical behavior was found to be non-normal and in many cases exhibited moderate to strong joint randomness.

3. What advances in probability and statistics theory have been made which show promise in overcoming such limitations?

   - Only a few mathematical models, such as the Nataf, Morgenstern and Bandte models, existed which could model joint randomness behavior. However, these

methods require initial statistical information of the random variables. In the case of local input parameters for a component reliability assessment, this information would not be available. An efficient single response variance approximation approach already existed in the literature. However, a multi-response covariance approximation technique does not. Additionally, several approaches for empirically identifying the appropriate parametric univariate distribution are available. These were found to be highly useful to the current study.

4. Are new advances in probability and statistics theory required?

- Yes, new advances are required as a thorough survey of the existing literature failed to produce a suitable framework of tools to model this complex statistical behavior. Two new methods have been developed within this study which have been shown to be highly accurate and efficient. In fact, they successfully capture joint, nonnormal randomness for a very complex turbine blade reliability analysis conducted within this study. These methods also permit the use of the canonical correlation analysis which is well-suited for reducing the complex statistical space of such multi-response, multi-analyses problems as that encountered in this study.

5. What would be an ideal framework of such advances and accepted methods?

- An ideal framework would be one that captures the complex statistical randomness between the system and the component of interest while doing so in an efficient manner. As demonstrated in this study, the following characteristics provide for a complete framework to meet the objectives of this research: 1) blending of physics-based models with appropriate probability and statistical tools, 2) interface to existing reliability methods, and 3) is constructed using multidisciplinary analysis elements so as to be amenable to design and optimization activities.

## 6.2 Summary of Contributions

The development of a suitable framework to enable the accurate, efficient component reliability assessment was the focus of this study. This framework is very unique in whole due to its systematic approach to addressing the difficult nature of the problem of conducting component reliability assessments. But, the framework as a whole is only one of several contributions created within this study. Given the requirements identified for the framework formulation, several new capabilities with the mathematical tools necessary to enable framework capabilities had to be created. They are now summarized.

- Multi-Response Joint Statistical Property Estimator

  - The most noteworthy contribution of the study is the creation of a multi-response joint statistical property estimator. This capability has been enabled through the derivation of efficient covariance approximation formulae which is an extension of the single response variance estimation technique found in conventional structural reliability theory. This mathematical contribution has been utilized several times for several different tasks within this study. One of the novel joint probability methods as well as the approximate canonical correlation analysis developed within this study require, as input, the results of the covariance approximation formulae.

- Novel Joint Probability Models

  - Two new joint probability models have been developed that are well-suited to the needs of the current study. Both models are unique compared to existing joint probability approaches in that they enable the generation of a joint probability space model of a vector of responses where only the response analysis(es) and statistical information of the response inputs is available. Existing joint

probability models require that the response distributions and correlation structure already exist. These new models were evaluated carefully for several cases, where they were found to be superior to even the existing models provided with accurate response distribution information. Moreover, they are more efficient than even the response surface Monte Carlo method in determining response parameter statistical properties. The multi-response first order second moment method, MFOSM, utilizes the covariance approximation technique in concert with the Anderson-Darling test statistics. The covariance approximation calculation within this method is easy to execute and can model joint (correlated) normal input behavior. But, it can not model non-normal input parameter behavior. The inverse probability model, IPDF, developed within this study overcomes this limitation. Similar to the MFOSM method IPDF also uses the first order sensitivity derivatives but does so to construct a linearized system of equations representing the multi-response space and is ideally suited for the case where input parameters are jointly distributed *and* non-normal. However, the IPDF method is less efficient than the MFOSM method when the input parameters are normally distributed.

- Multi-Response Parameter Space Reduction

- A new parameter space reduction technique has been demonstrated whereby multiple responses are considered simultaneously when deciding which parameters to focus on. The technique applies the canonical correlation analysis to rank order the importance of both vectors of parameters as well as individual parameters in terms of their contribution to all of the responses simultaneously. Where traditional canonical correlation uses large sample input, the input here is provided by the covariance approximation technique requiring only a limited

number of system evaluations. Screening exercises using the canonical correlation method is highly suited for components operating in complex systems because of the large number of analyses, responses, and intermediate and top-level inputs. Thus, this technique can directly consider much of the complexity associated with analyzing a complex system.

- Non-deterministic Analysis of FEA Responses

- A non-deterministic response calculation technique has been demonstrated where quantities such as FEA response variances and preliminary failure probability estimates can be made. Thus, an improved failure hot spot capability using statistical information is now available. This capability is also enabled using the covariance approximation technique developed in this study.

- Integrated Multi-disciplinary Component Failure Environment

- A one of a kind automated FEA-based component failure environment has been created to facilitate the research conducted within this study. The uniqueness of this environment is in the number and complexity of parameters that the environment can model. Full automation of the analysis required an initial investment of time and resources that was more than worthwhile once the many evaluations were conducted for this study.

## *6.3 Recommendations*

As with any study, several areas are to be identified where improvements and continued focus should be made. General areas of recommendations are considered including: 1) Design and Optimization implications, 2) Improved Joint Probability Modeling, and 3) Modeling Spatial and temporal randomness.

### 6.3.1 Design and Optimization

The design and optimization area is an important one as the analytical capability of this framework alone is only partially useful to the engineering process. By casting the framework into a design and optimization scheme, the information generated through the framework analysis can be brought into design and development activities to greatly improve the accuracy and efficiency of critical design decisions. A noteworthy feature of the framework is the use of vector-vector canonical correlation which can now provide a statistics-based parameter representing correlation between a vector of responses and the design input variables. Creating a design algorithm using this multi-response parameter could lead to finding design points in a more efficient and simplified manner. *Additionally, gradient-based design point search algorithms now can include joint statistical information using only the gradient information.* No additional evaluations over and above the sensitivity derivative would be required. This is highly useful as sensitivity derivatives are usually conducted for the conventional optimization method anyways. Also, approximate canonical correlation analysis (ACCA) can now provide additional insight to the rank order of any conceived combination of parameters at any level modelled in the analysis. For instance, *knowing which contributing analyses were stronger drivers of the response vector would allow for the intelligent control of the multi-disciplinary analysis space. Dominant contributing analyses and their driving input variable subsets can now be identified quickly to as to maximize the change in the objective function for each design step.* Another application of canonical correlation is the creation of a *model of models* where an applied statistical model of existing applied statistical models can now be created. For instance, for a given complex system, application of statistical modelling techniques result in numerous response models for several system and subsystem parameters. Using canonical correlation as a modelling technique can allow for the rapid creation of a model of these models that is unique to the vector candidate

objective functions and maximizes the fitness of the meta-metamodel with respect to the input variables. All of these activities can now be done analytically using the covariance approximation formulae derived during this research. Thus, statistical information of the input, intermediate, and response parameters can now be included. The current and future capabilities of the developed framework clearly show promise in improving the complex system design and optimization, especially in the presence of uncertainty.

### 6.3.2 Improved Joint Probability Modelling

The novel joint probability models created for this research are a major improvement to component reliability modelling. However, these models still have their limitations. One limitation shared between both of the new joint probability models is the use of a first order approach to approximate the response derivatives or construct a linear representation of what could actually be a non-linear space. Fortunately, good accuracy can be achieved with the first order approach as shown within this study even for complicated statistical randomness. But, additional accuracy and generality can be achieved by developing a second order approach. With a second order approach, second derivatives of the response parameters are available and a second order, quadratic Taylor series approximation of the actual system would be possible. With second order information, the covariance approximation technique should be able to account for higher order statistics of the response space such as skewness and kurtosis. By directly approximating these higher order statistics, distribution identification could be applied directly to the approximation results rather than requiring randomly generated evaluations. In the current first order rendition of the covariance approximation, a combined analytical, statistical approach is taken to identify the appropriate distribution and estimate its parameters. With a second order approach, a two-parameter distribution could be identified directly without the

need of randomly generated cases. This would prevent incorrectly specifying a distribution. In essence, distribution identification using covariance approximation would actually spawn a new sub-field within probability and statistics where new similarity parameters between approximate and actual statistics would be required.

For the inverse transformation model, a second order approach would require a tremendous amount of effort to create. Analytically this would not be difficult as a second order Taylor series approximation of the system could be implemented and inverse solution formulae exist for quadratic systems. However, the difficulty would be in performing necessary integration of the resulting joint density function where the integration space is non-unique, a property of quadratic inverse solutions. Therefore, unless applied mathematical models to enable multi-dimensional, non-unique integration of quadratic spaces exist this modification is not recommended.

The joint covariate model proposed in section 3.4.3 is an excited idea as it can combine joint probability models in addition to statistically parameterized covariate models. Thus, multiple responses would be permissible within this mathematical proposition. Two additional improvements to this idea are recommended. The first improvement is a necessary one and entails creating the ability to account for nonconstant correlation behavior. The existing joint covariate model relies on the assumption that the correlation structure remains constant for different values of the covariate (input) variable values. This assumption can break down when the correlation structure differs greatly across the parameterized covariate space. A modification to the general covariate model is recommended where an additional parameter(s) is included that can account for varying correlation characteristics. This implies a modification to the maximum-likelihood method used in calculating the covariate model coefficients. The second modification to the joint covariate model involves translating it from a statistical-sample based model to an analytical model where the covariance approximation formulae can be utilized. This is a more radical approach but potential

can directly account for the change in the response correlation structure.

Additional recommendations are to be made involving the linkage between the framework's joint probability models and other advanced probability and statistics topics. As the joint probability and statistical data is generated through computational simulation within this study, the ability to incorporate experimental, production, and operational data from actual physical components is recommended. A straightforward means of incorporating this data is to use the popular Bayesian updating approach. Conceptually, this would probability involve using the computational framework to specify the prior distribution and applying the new data, from any given source, to create the posterior distribution for improved reliability predictions. A literature review is recommended to see if Bayesian theory includes multi-variate joint probability principles. Another advance topic is that of non-parametric statistics. During the implementation of the formulated framework it was found through statistical hypothesis testing that none of the candidate parametric probability distributions could, with significance, be shown to follow the cycle analysis responses. Although the framework recommended and implemented the lognormal distribution which achieved a very reasonable result, the remaining 3% error could perhaps be removed by better modelling the actual cycle response distributions. One way to do this is to use non-parametric methods. These methods are free from the assumption of a parametric distribution, but only work with the empirical distribution data. Storage and interpretation of the necessary data might prove to be a deterrent. Perhaps, a more flexible family of parametric joint probability distributions can be created and thus be readily used within the formulated framework. Additional consideration of this topic is recommended.

### 6.3.3 Modelling Spatial and Temporal Randomness

Two entire fields are worthy of mentioning here for future work to be considered. They are stochastic process and random fields. Stochastic processes are essentially random variables of the time domain where random fields are random variables of the spatial domain. The random variables considered in this study were singular and were assumed to be statistically time and spatially invariant. Some applications, even gas turbine components, have been shown to exhibit temporal and spatial randomness. And, researchers have shown that neglecting such behavior also can confound the analysts ability to predict accurate component reliability values. Where the previous recommendations are natural extensions to the existing body of work pursued here, random field and stochastic processes would require a major effort. Modelling this phenomena requires a new approach mathematically and greatly increases the dimensionality of the problem. Nonetheless, the basic formulation of the developed framework in this study already includes the necessary elements for modelling stochastic and even spatially random fields as they are in fact basically advanced joint probability distributions.

# APPENDIX A

# COVARIANCE FORMULAE

## *A.1 Single Response Variance Approximation*

The derivation of the first order approximation of a response variation is given. To begin with, either a linear relationship is known or found using some approximation technique where the response is represented by a linear function of several input variables of the form

$$y = a_o + a_i + \ldots + a_n x_n = a_o + \sum_{i=1}^{n} a_i x_i \tag{93}$$

where the coefficients $a_o$ and $a_i$, if not specified, can be approximated for an unknown function using numerical differentiation. The mean value of the response is then given as[1]

$$E[y] = E\left[a_o + \sum_{i=1}^{n} a_i x_i\right] = a_o + \sum_{i=1}^{n} a_i E[x_i] = a_o + \sum_{i=1}^{n} a_i \mu_{x_i} \tag{94}$$

Then, by the definition of covariance

$$VAR(y) = COV(y, y) \tag{95}$$

and substituting equation (93) into equation (95) becomes

$$COV(y, y) = COV\left(\left[a_o + \sum_{i=1}^{n} a_i x_i\right], \left[a_o + \sum_{j=1}^{n} a_j x_j\right]\right)$$

[1]If the function is unknown, then equation (94) is an approximation to the mean of the actual function.

Next, by the properties of Covariance of linear sums of variables the following simplifications can be made

$$COV\left(\left[a_o+\sum_{i=1}^{n}a_ix_i\right],\left[a_o+\sum_{j=1}^{n}a_jx_j\right]\right)=COV\left(a_o,a_o\right)+COV\left(a_o,\left[\sum_{j=1}^{n}a_jx_j\right]\right)$$
$$+COV\left(\left[\sum_{i=1}^{n}a_ix_i\right],a_o\right)+COV\left(\left[\sum_{i=1}^{n}a_ix_i\right],\left[\sum_{j=1}^{n}a_jx_j\right]\right) \tag{96}$$

Since the variance of a constant or the covariance of a constant and a random variable is equal to zero, the first three terms of equation (96) can be dropped giving the following result

$$COV\left(\left[a_o+\sum_{i=1}^{n}a_ix_i\right],\left[a_o+\sum_{j=1}^{n}a_jx_j\right]\right)=COV\left(\left[\sum_{i=1}^{n}a_ix_i\right],\left[\sum_{j=1}^{n}a_jx_j\right]\right) \tag{97}$$

where for an unknown function $a_i=\frac{\partial y}{\partial x_i}$. The same property applied to produce equation (96), can be applied to simplify equation (97) as follows

$$COV\left(\left[a_o+\sum_{i=1}^{n}a_ix_i\right],\left[a_o+\sum_{j=1}^{n}a_jx_j\right]\right)=\sum_{i=1}^{n}\sum_{j=1}^{n}a_ia_jCOV(x_i,x_j) \tag{98}$$

Using the result of equation (98) and substituting $a_i\approx\frac{\partial y}{\partial x_i}$, the single response variance can then be expressed as

$$VAR(y)\approx\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j}COV(x_i,x_j) \tag{99}$$

## *A.2 Multi-Response Covariance Approximation*

The objectives of this research study require the efficient quantification of the joint randomness of several response functions and therefore equation (99) must be generalized to provide the covariance of a pair of responses. Using the covariance of a pair of responses one can then construct the entire covariance matrix of a system of responses. The derivation of the covariance approximation is similar to that of the variance approximation.

First, define a system of responses whose linear approximations are given as

$$\begin{aligned} y_1 &= a_{1o} + \sum_{i=1}^{n} a_{1i}x_i \\ y_2 &= a_{2o} + \sum_{i=1}^{n} a_{2i}x_i \\ &\vdots \\ y_m &= a_{mo} + \sum_{i=1}^{n} a_{mi}x_i \end{aligned} \tag{100}$$

or in matrix form

$$\mathbf{y}_{m\times 1} = \mathbf{M}_{m\times n}\mathbf{x}_{n\times 1}$$

The covariance then between the $i^{th}$ and $j^{th}$ response using equation (100) is given as

$$COV(y_i, y_j) = COV\left(\left[a_{io} + \sum_{k=1}^{n} a_{ik}x_k\right], \left[a_{jo} + \sum_{l=1}^{n} a_{jl}x_l\right]\right) \tag{101}$$

Again, by the properties of Covariance

$$\begin{aligned} &COV(y_i, y_j) = COV\left(a_{io}, a_{jo}\right) + COV\left(a_{io}, \left[\sum_{l=1}^{n} a_{jl}x_l\right]\right) + \\ &COV\left(\left[\sum_{k=1}^{n} a_{ik}x_k\right], a_o\right) + COV\left(\left[\sum_{k=1}^{n} a_{ik}x_k\right], \left[\sum_{l=1}^{n} a_{jl}x_l\right]\right) \end{aligned} \tag{102}$$

Since the variance of a constant or the covariance of a constant and a random variable is equal to zero, the first three terms of equation (102) can be dropped giving the following result

$$COV\left(\left[a_{io} + \sum_{m=1}^{n} a_{im}x_m\right], \left[a_{jo} + \sum_{l=1}^{n} a_{jl}x_{jl}\right]\right) = COV\left(\left[\sum_{m=1}^{n} a_{im}x_m\right], \left[\sum_{l=1}^{n} a_{jl}x_l\right]\right) \tag{103}$$

where for an unknown function $a_i = \frac{\partial y}{\partial x_i}$. The same property applied to produce equation (102), can be applied to simplify equation (103) as follows

$$COV\left(\left[a_{io} + \sum_{m=1}^{n} a_{im}x_{im}\right], \left[a_{jo} + \sum_{l=1}^{n} a_{jl}x_{jl}\right]\right) = \sum_{m=1}^{n}\sum_{l=1}^{n} a_{im}a_{jl}COV(x_m, x_l) \tag{104}$$

Using the result of equation (104) and substituting $a_{im} \approx \frac{\partial y_i}{\partial x_m}$, the approximation of the covariance between the $i^{th}$ and $j^{th}$ responses can then be expressed as

$$COV(y_i, y_j) \approx \sum_{m=1}^{n} \sum_{l=1}^{n} \frac{\partial y_i}{\partial x_m} \frac{\partial y_j}{\partial x_l} COV(x_m, x_l) \tag{105}$$

## *A.3 Response-Input Covariance Approximation*

The approximation of the covariance between the responses and the input variables can also be derived similar to sections A.1 and A.2. Beginning with the definition of the linear system of response equations represented by equation (100), the covariance between the $i^{th}$ response and the $j^{th}$ input variable can be expressed as

$$COV(y_i, x_j) = COV\left(\left[a_{io} + \sum_{k=1}^{n} a_{ik} x_k\right], x_j\right) \tag{106}$$

which by application of the properties of covariance can be expressed as

$$COV\left(\left[a_{io} + \sum_{k=1}^{n} a_{ik} x_k\right], x_j\right) = COV\left(a_{io}, x_j\right) + COV\left(\left[\sum_{k=1}^{n} x_k\right], x_j\right) \tag{107}$$

Since $a_{io}$ is a constant, equation (107) reduces to

$$COV\left(\left[a_{io} + \sum_{k=1}^{n} a_{ik} x_k\right], x_j\right) = COV\left(\sum_{k=1}^{n} a_{ik} x_k, x_j\right) \tag{108}$$

Applying the properties of covariance again produces the following more compact form

$$COV\left(\left[\sum_{k=1}^{n} a_{ik} x_k\right], x_j\right) = \sum_{k=1}^{n} a_{ik} COV\left(x_k, x_j\right) \tag{109}$$

Substituting $a_{ik} \approx \frac{\partial y_i}{\partial x_k}$ as the coefficient approximation for an unknown function, the formula for approximating the covariance between the $i^{th}$ response and the $j^{th}$ input variable can be expressed as

$$COV\left(y_i, x_j\right) \approx \sum_{k=1}^{n} \frac{\partial y_i}{\partial x_k} COV\left(x_k, x_j\right) \tag{110}$$

Finally, by using the symmetric property of the covariance function the covariance of the $i^{th}$ input variable and the $j^{th}$ response would be approximated as

$$COV\left(x_i, y_j\right) \approx \sum_{k=1}^{n} \frac{\partial y_j}{\partial x_k} COV\left(x_i, x_k\right) \tag{111}$$

Thus, using equations (110) and (111) one can solve for the entire covariance matrix between each response-input pair. Note that only one of these two equations is necessary since the covariance matrix is symmetric.

# APPENDIX B

# JOINT PROBABILITY PROGRAMS

## *B.1 Bar Reliability Example*

demandresistance.m

```
% JMW Thesis
% 10/13/03
%
% Simple fixed-fixed beam example to demonstrate joint probability importance
%
% Input Constants
%   Rupture stress formula (resistance) R=a*T+b where T is temperature and
%   a,b are constants (see Moon et al. 1968)
a=-0.1695;      % Rupture stress temperature coefficient
b=329;          % Rupture stress temperature intercept
%   Thermal induced stress (demand) formula input   D=E*alpha*(T-Tref)
%   where E is the elastic modulus, alpha is the thermal expansion
%   coefficient, and Tref is the reference temperature and is a constant
%
% Material Properties
alpha=8.6e-6;  % Thermal expansion coefficient
E=31000;      % Reference temperature for determining temperature change (
%
% Temperature variables statistics
muTr=1800;  % mean clamping reference temperature
sigTr=10;   % standard deviation of clamping reference temperature
muT=1700;   % mean temperature
sigT=10;    % standard deviation of temperature
%
% Dependence solution
muG=a*muT+b+E*alpha*(muT-muTr); % Expected value (mean) of g-function
sigR=(a^2*sigT^2)^0.5;
sigD=((E*alpha)^2*sigT^2+(E*alpha)^2*sigTr^2)^0.5;
rhoDR=-(a*E*alpha*sigT^2)/sigR/sigD;     % Correlation coefficient between Demand and Resistance
covDR=-a*E*alpha*sigT^2;
sigG=(sigR^2+sigD^2-2*covDR)^0.5;
pfdep=normcdf(-muG/sigG,0,1);   % Compute failure probability using standardized
                                % g-function normal random variable
% Independence solution
muR=a*muT+b;
muD=-E*alpha*(muT-muTr);
mug=muR-muD;
sigg=(sigR^2+sigD^2)^0.5;
pfind=normcdf(-mug/sigg,0,1);
%
% g-fcn plots
n=1E5;
g1=random('Normal',mug,sigg,n,1);
g1=sort(g1);
g1p=normpdf(g1,mug,sigg);
g2=random('Normal',mug,sigG,n,1);
g2=sort(g2);
g2p=normpdf(g2,mug,sigG);
plot(g1,g1p,'-.',g2,g2p,'-')
xlabel('Stress (ksi)');
ylabel('Probability Density');
legend('Independent Solution','Actual Solution');
legend('Independent Solution','Actual Solution',0);
```

## *B.2 Linear System with Joint Normal Input*

### B.2.1 EDF Method

edfnorm.m

```
% Thesis Proposal Study
% Pedagogical Example (April 13, 2003)
%
% ------------------
% Independent input
% ------------------
```

```
N=1E6;
mu=20;
sig=2;
maxy1=46.5;
maxy2=70;
x1=random('norm',mu,sig,N,1);
x2=random('norm',mu,sig,N,1);
% Visualization of input distribution
figure
[rho,xv1,xv2]=density(x1,x2,-100,-100,[14 26 14 26]);
[xv1,xv2]=meshgrid(xv1,xv2);
mesh(xv1,xv2,rho)
axis([14 26 14 26 0 0.05])
xlabel('x1')
ylabel('x2')
zlabel('density')
figure
[rho,xv1,xv2]=density(x1,x2,-30,-30,[14 26 14 26]);
contour(xv1,xv2,rho)
axis([14 26 14 26])
xlabel('x1')
ylabel('x2')
%
y1=x1+x2;
y2=x1+2*x2;
clear x1 x2
% Visualization of output distribution
figure
[rho,yv1,yv2]=density(y1,y2,-100,-100,[30 50 45 75]);
[yv1,yv2]=meshgrid(yv1,yv2);
mesh(yv1,yv2,rho)
axis([30 50 45 75 0 0.05])
xlabel('y1')
ylabel('y2')
zlabel('density')
figure
[rho,yv1,yv2]=density(y1,y2,-60,-60,[30 50 45 75]);
contour(yv1,yv2,rho,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
%
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
y=[y1 y2];
clear y1 y2
C=[0 0];
for j=[1 2];
    for i=[1:N];
        if j==1;
            limit=maxy1;
        else
            limit=maxy2;
        end
        if y(i,j)>=limit
        C(j)=C(j)+1;
        end
    end
end
Pfis=1-(1-C(1)/N)*(1-C(2)/N)        % Series event (ind) probability formula
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:N];
        if y(i,1)>=maxy1 | y(i,2)>=maxy2
        C=C+1;
        end
end
clear y
Pfid=C/N        % Joint Probability
%
%
%
%
% ------------------------------------------------
% Dependent input: Bivariate normal distribution
% ------------------------------------------------
p=0.5;  % Correlation coefficient of x1, x2 pair
U=random('norm',0,1,N,1); % Must be std normal RV for formula
V=random('norm',0,1,N,1); % Must be std normal RV for formula
x1=mu+sig*U;
x2=mu+sig*p*U+sig*((1-p^2)^0.5)*V;
clear U V
% Visualization of input distribution
figure
[rho,xv1,xv2]=density(x1,x2,-100,-100,[14 26 14 26]);
[xv1,xv2]=meshgrid(xv1,xv2);
mesh(xv1,xv2,rho)
axis([14 26 14 26 0 0.05])
xlabel('x1')
ylabel('x2')
zlabel('density')
figure
[rho,xv1,xv2]=density(x1,x2,-30,-30,[14 26 14 26]);
```

```
contour(xv1,xv2,rho)
axis([14 26 14 26])
xlabel('x1')
ylabel('x2')
%
y1=x1+x2;
y2=x1+2*x2;
clear x1 x2
y=[y1 y2];
rhoy1y2=corrcoef(y)  % rho=0.9488
muV=[mean(y1) mean(y2)];    % muV=[39.99772962495611 59.99770173384864]
sigV=[std(y1) std(y2)];
% Visualization of output distribution
figure
[rho,yv1,yv2]=density(y1,y2,-100,-100,[30 50 45 75]);
[yv1,yv2]=meshgrid(yv1,yv2);
mesh(yv1,yv2,rho)
axis([30 50 45 75 0 0.05])
xlabel('y1')
ylabel('y2')
zlabel('density')
figure
[rho,yv1,yv2]=density(y1,y2,-60,-60,[30 50 45 75]);
contour(yv1,yv2,rho,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
%
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
y=[y1 y2];
clear y1 y2
C=[0 0];
for j=[1 2];
    for i=[1:N];
        if j==1;
            limit=maxy1;
        else
            limit=maxy2;
        end
        if y(i,j)>=limit
        C(j)=C(j)+1;
        end
    end
end
Pfcs=1-(1-C(1)/N)*(1-C(2)/N)        % Series event (ind) probability formula
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:N];
        if y(i,1)>=maxy1 | y(i,2)>=maxy2
        C=C+1;
        end
end
%clear y
Pfcd=C/N        % Joint probability
%
% Semi-Parametric Distribution Function Method (Bandte 2000)
%
rho=corrcoef(y)
fy1=6621238954613787/144115188075855872*exp(-1/24*y1^2+10/3*y1-200/3)*2^(1/2)*pi^(1/2); % Determined
                                                                                        % using INV solution
fy2=945891279230541/144115188075855872*exp(-1/56*y2^2-450/7+15/7*y2)*7^(1/2)*6^(1/2)*pi^(1/2);
                                                                                % Determined using INV solution
```

## B.2.2 IPDF Method

$IPDF_case1.m$

```
% INV Solution to Derive and Solve for SPDF input
mu=20;
sig=2;
maxy1=46.5;
maxy2=70;
syms x1 x2 y1 y2
p=0.5;  % Correlation coefficient of x1, x2 pair
fx1x2=((2*pi*sig^2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((x1-mu)/sig)^2)-2*p*((x1-mu)/sig)...
*((x2-mu)/sig)+((x2-mu)/sig)^2))
fy1y2=subs(subs(fx1x2,x1,2*y1-y2),x2,-y1+y2)
% Support of Chapter IV: Compute Analytical Correlation Coefficient
    % NOTE: Symbolic solutions can be found for each of these equations
fy1=int(fy1y2,y2,-inf,+inf);    % Solve for y1 marginal distribution
fy2=int(fy1y2,y1,-inf,+inf);    % Solve for y2 marginal distribution
Efy1=numeric(int(fy1*y1,-inf,inf));      % Solve for expected value of y1
Efy2=numeric(int(fy2*y2,-inf,inf));      % Solve for expected value of y2
stdy1=numeric((int((y1-Efy1)^2*fy1,y1,-inf,inf))^0.5); % Solve for std. dev. of y1
stdy2=numeric((int((y2-Efy2)^2*fy2,y2,-inf,inf))^0.5); % Solve for std. dev. of y2
corry1y2=numeric(int(int(((y1-Efy1)/stdy1)*((y2-Efy2)/stdy2)*fy1y2,y1,-inf,inf),y2,-inf,inf));
% Visualization of IPDF solution
s=3;    % Number of standard deviations for plotting
yy1=linspace(Efy1-s*stdy1,Efy1+s*stdy1,100);
```

```
yy2=linspace(Efy2-s*stdy2,Efy2+s*stdy2,100);
for i=[1:length(yy1)]
    for j=[1:length(yy2)]
    fy1y2I(i,j)=subs(subs(fy1y2,y1,yy1(i)),y2,yy2(j));
    end
end
%mesh(fy1y2N)
contour(yy1,yy2,fy1y2I,10)
axis([(Efy1-s*stdy1), (Efy1+s*stdy1), (Efy2-s*stdy2), (Efy2+s*stdy2)])
xlabel('y1')
ylabel('y2')
% Visualize actual g-function   fy1y2=fy1*fy2*g solve for g
gfy1y2=fy1y2/(fy1*fy2);
for i=[1:length(yy1)]
    for j=[1:length(yy2)]
    gfy1y2I(i,j)=36028797018963968/19863716863841361*exp(-2/3*(yy1(i)-1/2*yy2(j)-10)^2+2/3*...
    (yy1(i)-1/2*yy2(j)-10)*(-1/2*yy1(i)+1/2*yy2(j)-10)-2/3*(-1/2*yy1(i)+1/2*yy2(j)-10)^2)/...
    exp(-1/24*yy1(i)^2+10/3*yy1(i)-200/3)*2^(1/2)/pi/exp(-1/56*yy2(j)^2-450/7+15/7*yy2(j))*...
    7^(1/2)*6^(1/2);
    end
end
[yy1,yy2]=meshgrid(yy1,yy2);
mesh(yy1,yy2,gfy1y2I)
%axis([(Efy1-s*stdy1), (Efy1+s*stdy1), (Efy2-s*stdy2), (Efy2+s*stdy2)])
xlabel('y1')
ylabel('y2')
zlabel('g-function value')
%
% BPDF Method
r=20;    % Number of standard deviations
y1max=numeric(Efy1+r*stdy1);  % Calculate ymax using r*stdy1
y1min=numeric(Efy1-r*stdy1);  % Calculate ymin using r*stdy1
y2max=numeric(Efy2+r*stdy2);  % Calculate ymax using r*stdy2
y2min=numeric(Efy2-r*stdy2);  % Calculate ymin using r*stdy2
y1med=numeric((y1max+y1min)/2); % Calculate y1 median
y2med=numeric((y2max+y2min)/2); % Calculate y2 median
y1range=numeric((y1max-y1min)/2); % Calculate y1 range
y2range=numeric((y2max-y2min)/2); % Calculate y2 range
g=1+corry1y2*((y1-y1med)/y1range)*((y2-y2med)/y2range)
% Plot g function
ezmesh(g,[y1min y1max y2min y2max],50)
% Plot g/C function
C=int(int(fy1*fy2*g,y1,-inf,inf),y2,-inf,inf);
gC=g/C;
ezmesh(gC,[y1min y1max y2min y2max],50)
ezmesh(ginvsol,[y1min y1max y2min y2max],500)
% Solve for failure probability using SPDF
fy1y2SPDF=(1/C)*fy1*fy2*g
fy1y2SPDF=simple(simplify(fy1y2SPDF));
pfuinv=1-numeric(int(int(fy1y2SPDF,y1,y1min,maxy1),y2,y2min,maxy2))
% Construct g/C using INV method
%ginvsolC=fy1y2/(fy1*fy2);       % solve for g/C ratio using inverse transformation solution
%C=int(int(fy1*fy2*ginvsolC,y1,-inf,inf),y2,-inf,inf);
%ezmesh(ginvsolC,[y1min y1max y2min y2max],500)
% Solve for failure probability using IPDF
pfuinv=1-numeric(int(int(fy1y2,y1,y1min,maxy1),y2,y2min,maxy2))
%
% Parametric Representation
% INV Solution to Derive and Solve for SPDF input
syms x1 x2 y1 y2 mu sig a b c d p
%p=0.5;  % Correlation coefficient of x1, x2 pair
fx1x2=((2*pi*sig^2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((x1-mu)/sig)^2)-2*p*((x1-mu)/sig)*...
((x2-mu)/sig)+((x2-mu)/sig)^2))
fy1y2=subs(subs(fx1x2,x1,a*y1+b*y2),x2,c*y1+d*y2)
% Support of Chapter IV: Compute Analytical Correlation Coefficient
% NOTE: Symbolic solutions can be found for each of these equations, but numeric used for reporting
fy1=int(fy1y2,y2,-inf,+inf);    % Solve for y1 marginal distribution
fy2=int(fy1y2,y1,-inf,+inf);    % Solve for y2 marginal distribution
```

### B.2.3 NPDF Method

$NPDF_{case1.m}$

```
% JMW Thesis Chapter: Joint Probability Models
%
% NPDF_case1.m is the implementation of various transformation approaches for
% comparison to IPDF and BPDF
%
% ------------------------------------------------------------
% Sub case 1:  Find JPDF of responses and proceed with NPDF
% ------------------------------------------------------------
%
% INV Solution to Derive and Solve for NPDF input
mu=20;
sig=2;
maxy1=46.5;
maxy2=70;
```

```
syms x1 x2 y1 y2
p=0.0;  % Correlation coefficient of x1, x2 pair
fx1x2=((2*pi*sig^2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((x1-mu)/sig)^2)-2*p*((x1-mu)/sig)*...
((x2-mu)/sig)+((x2-mu)/sig)^2))
fy1y2=subs(subs(fx1x2,x1,2*y1-y2),x2,-y1+y2)
% Support of Chapter IV: Compute Analytical Correlation Coefficient
    % NOTE: Symbolic solutions can be found for each of these equations
fy1=int(fy1y2,y2,-inf,+inf);    % Solve for y1 marginal distribution
fy2=int(fy1y2,y1,-inf,+inf);    % Solve for y2 marginal distribution
Efy1=numeric(int(fy1*y1,-inf,inf));      % Solve for expected value of y1
Efy2=numeric(int(fy2*y2,-inf,inf));      % Solve for expected value of y2
stdy1=numeric((int((y1-Efy1)^2*fy1,y1,-inf,inf))^0.5); % Solve for std. dev. of y1
stdy2=numeric((int((y2-Efy2)^2*fy2,y2,-inf,inf))^0.5); % Solve for std. dev. of y2
corry1y2=numeric(int(int(((y1-Efy1)/stdy1)*((y2-Efy2)/stdy2)*fy1y2,y1,-inf,inf),y2,-inf,inf));
R=[1 corry1y2; corry1y2 1];
%
% TPDF Method 1: Nataf Method
syms z1 z2
% Use response mean and variance vectors as well as correlation coefficient to find R'(rho12')
%   Must use a numerical method function call to find rho12p given these inputs
muV=[Efy1 Efy2];        % Vector of response means (marginal distribution)
sigV=[stdy1 stdy2];     % Vector of response standard deviations (marginal distribution)
% Create joint standard normal dividend which introduces joint randomness in NATAF model
Fy1=int(fy1,y1)+0.5;    % y1 marginal cumulative dist
Fy2=int(fy2,y2)+0.5;     % y2 marginal cumulative dist
Finvy1=finverse(Fy1);    % Find inverse function of y1 CDF
Finvy2=finverse(Fy2);    % Find inverse function of y2 CDF
% Solution routine for the joint NATAF model parameters (rhop)
n=2;    % Specify the number of variables
s=7;    % Numer of standard deviations for integration range (must not be too large or small)

for i=[1:n-1]
    for j=[i+1:n]
        rho=R(i,j);
        muy1=muV(1);
        muy2=muV(2);
        rhoo=dblquad(@drhoij,-5,5,-5,5,1e-6,[],stdy1,stdy2,muy1,muy2,rho,Finvy1,Finvy2)
        R(i,j)-int(int(
    end
end
rhoarray=[0.93:0.0005:0.97];
for i=[1:length(rhoarray)]
    rho=rhoarray(i);
    rhoo(i)=dblquad(@drhoij,-5,5,-5,5,1e-6,[],stdy1,stdy2,muy1,muy2,rho,Finvy1,Finvy2);
end
%
%
%
%-------------------------------------------------------------------------------------------
% Solve Case 1 (Dependent Input) using NATAF model (Numerical Integration)
% ---------------------------------------------------------
rhop=0.9487;     % Determined through iterative numerical integration
s=7;
pfunataf=dblquad(@fnataf,Efy1-s*stdy1,maxy1,Efy2-s*stdy2,maxy2,1e-6,[],fy1,fy2,Fy1,Fy2,rhop);
% Visualize Nataf joint random term (phi)
clear yy1 yy2
yy1=linspace(30,50,50);
yy2=linspace(45,75,50);
for i=[1:length(yy1)]
    for j=[1:length(yy2)]
    fy1a=numeric(6621238954613787/144115188075855872*exp(-1/24.*yy1(i).^2+10/3.*yy1(i)-200/3)...
    .*2^(1/2)*pi^(1/2));
    fy2a=numeric(945891279230541/144115188075855872*exp(-1/56.*yy2(j).^2-450/7+15/7.*yy2(j))...
    .*7^(1/2)*6^(1/2)*pi^(1/2));
    Fy1a=6621238954613787/144115188075855872*2^(1/2)*pi*6^(1/2)*erf(1/12*6^(1/2).*yy1(i)-10/3*...
    6^(1/2))+1/2;
    Fy2a=945891279230541/144115188075855872*7^(1/2)*6^(1/2)*pi*14^(1/2)*erf(1/28*14^(1/2).*yy2(j)...
    -15/7*14^(1/2))+1/2;
    z1=norminv(Fy1a);
    z2=norminv(Fy2a);
    phi(i,j)=((2*pi*(1-rhop^2)^0.5)^-1)*exp(((2*rhop^2-2)^-1)*(z1.^2-2*rhop.*z1.*z2+z2.^2))./...
    (normpdf(z1).*normpdf(z2));
    fy1y2N(i,j)=fy1a*fy2a*phi(i,j);
    end
end
mesh(phi)
mesh(fy1y2N)
contour(yy1,yy2,phi,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
zlabel('density')
%
%-------------------------------------------------------------------------------------------
% Solve Case 1 (Independent Input) using NATAF model (Numerical Integration)
% ---------------------------------------------------------
rho=0.9487;     % Determined through iterative numerical integration
           % NOTE: rhop from NATAF and rho from original space are equivalent!
s=7;
pfunataf=dblquad(@fnataf,Efy1-s*stdy1,maxy1,Efy2-s*stdy2,maxy2,1e-6,[],fy1,fy2,Fy1,Fy2,rho);
%
```

```
%  Visualization of NPDF Solution
rho=0.9843;
yy1=linspace(30,50,50);
yy2=linspace(45,75,50);
[yy1,yy2]=meshgrid(yy1,yy2);
    fy1a=5734161139222659/72057594037927936*exp(-1/16.*yy1.^2+5.*yy1-100)*pi^(1/2);
    fy2a=5734161139222659/360287970189639680*exp(-1/40.*yy2.^2-90+3.*yy2)*5^(1/2)*2^(1/2)*pi^(1/2);
    Fy1a=5734161139222659/36028797018963968*pi*erf(1/4*yy1-10)+1/2;
    Fy2a=5734161139222659/360287970189639680*5^(1/2)*2^(1/2)*pi*10^(1/2)*erf(1/20*10^(1/2).*yy2...
    -3*10^(1/2))+1/2;
    z1=norminv(Fy1a);
    z2=norminv(Fy2a);
    phi=((2*pi*(1-rho^2)^0.5)^-1)*exp(((2*rho^2-2)^-1)*(z1.^2-2*rho.*z1.*z2+z2.^2))./((normpdf(z1)...
    .*normpdf(z2)));
    fy1y2N=phi*fy1a*fy2a;
mesh(yy1,yy2,fy1y2N)
%mesh(yy1,yy2,phi)
clear yy1 yy2
yy1=linspace(30,50,50);
yy2=linspace(45,75,50);
for i=[1:length(yy1)]
    for j=[1:length(yy2)]
    fy1a=5734161139222659/72057594037927936*exp(-1/16.*yy1(i).^2+5.*yy1(i)-100)*pi^(1/2);
    fy2a=5734161139222659/360287970189639680*exp(-1/40.*yy2(j).^2-90+3.*yy2(j))*5^(1/2)*2^(1/2)*...
    pi^(1/2);
    Fy1a=5734161139222659/36028797018963968*pi*erf(1/4*yy1(i)-10)+1/2;
    Fy2a=5734161139222659/360287970189639680*5^(1/2)*2^(1/2)*pi*10^(1/2)*erf(1/20*10^(1/2).*yy2(j)...
    -3*10^(1/2))+1/2;
    z1=norminv(Fy1a);
    z2=norminv(Fy2a);
    phi=((2*pi*(1-rho^2)^0.5)^-1)*exp(((2*rho^2-2)^-1)*(z1.^2-2*rho.*z1.*z2+z2.^2))./((normpdf(z1)...
    .*normpdf(z2)));
    fy1y2N(i,j)=phi*fy1a*fy2a;
    end
end
mesh(fy1y2N)
contour(yy1,yy2,fy1y2N,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
%
% ------------------------------------------------------------------------------------
% Validation case (Der Kiureghian 1986 example 2)
% ------------------------------------------------------------
%
muy1=1;
muy2=1;
stdy1=1;
stdy2=1;
rho=0.3;
fy1=exp(-y1);
fy2=exp(-y2);
rhoo=dblquad(@drhoijval,-1E8,1E8,-1E8,1E8,1e-6,[],stdy1,stdy2,muy1,muy2,rho)
```

### B.2.4 MFOSM Method

$MFOSM_case1_2.m$

```
% JMW Thesis   8/26/03
%
% Taylor Series Finite Difference Joint Probability Method (TPDF)
% for case 2:  Linear system of equations with dependent normal input
mu=20;
sig=2;
p=0.5;
maxy1=46.5;
maxy2=70;
mux=[mu mu];    % Define input vector of variable mean values (1xn)
muy=linearfcn(mux);  % Calculate mean of responses using linearfcn function
covx=[sig*sig p*sig*sig; p*sig*sig sig*sig];     % Define input covariance vector (n by n)
% Find input correlation coefficient matrix
D=sqrt(diag(diag(covx)));   % Find the square root of the diagonal of corrx
corrx=D^-1*covx*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Sensitivity Analysis (see equation 71)
ny=2;           % Provide number of responses
for i=[1:ny]
    for j=[1:length(mux)]
        xp=mux;
        xp(j)=xp(j)+covx(j,j);
        xm=mux;
        xm(j)=xm(j)-covx(j,j);
        yp=linearfcn(xp);
        ym=linearfcn(xm);
        dydx(i,j)=(yp(i)-ym(i))/(2*covx(j,j));
%        dydx(i,j)=(muy(i)-ym(i))/covx(j,j);
    end
end
% Response Covariance Matrix (COV) Approximation
```

```
covy=zeros(length(muy));
for i=[1:length(muy)]
    for j=[1:length(muy)]
        for m=[1:length(mux)]
            for n=[1:length(mux)]
                dcov(m,n)=dydx(i,m)*dydx(j,n)*covx(m,n);
            end
        end
        covy(i,j)=sum(sum(dcov));
    end
end
% Compute Correlation (CORR) matrix calculation using COV
D=sqrt(diag(diag(covy)));   % Find the square root of the diagonal of COVY
corry=D^-1*covy*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Construct Joint Normal Probability Space with TSFD results
syms y1 y2
sigy1=covy(1,1)^0.5
sigy2=covy(2,2)^0.5
p=corry(1,2);
fy1y2=((2*pi*sigy1*sigy2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((y1-muy(1))/sigy1)^2)-2*p*...
((y1-muy(1))/sigy1)*((y2-muy(2))/sigy2)+((y2-muy(2))/sigy2)^2));
% Visualize joint probability space
s=3;
ezsurf(fy1y2,[muy(1)-s*sigy1 muy(1)+s*sigy1 muy(2)-s*sigy2 muy(2)+s*sigy2],85)
%
% Compute failure probability
s=7;
pfuinv=1-dblquad(@finvsol,muy(1)-s*sigy1,maxy1,muy(2)-s*sigy2,maxy2,1e-6);
% pf=0.03497 (dependent input)
%%%
%
%
%
% Taylor Series Finite Difference Joint Probability Method (TPDF)
% for case 2:  Linear system of equations with independent lognormal input
mu=20;
sig=2;
p=0;
maxy1=46.5;
maxy2=70;
mux=[mu mu];    % Define input vector of variable mean values (1xn)
muy=linearfcn(mux);  % Calculate mean of responses using linearfcn function
covx=[sig*sig p*sig*sig; p*sig*sig sig*sig];     % Define input covariance vector (n by n)
% Find input correlation coefficient matrix
D=sqrt(diag(diag(covx)));   % Find the square root of the diagonal of corrx
corrx=D^-1*covx*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Sensitivity Analysis (see equation 71)
ny=2;           % Provide number of responses
for i=[1:ny]
    for j=[1:length(mux)]
        xp=mux;
        xp(j)=xp(j)+covx(j,j);
        xm=mux;
        xm(j)=xm(j)-covx(j,j);
        yp=linearfcn(xp);
        ym=linearfcn(xm);
        dydx(i,j)=(yp(i)-ym(i))/(2*covx(j,j));
%        dydx(i,j)=(muy(i)-ym(i))/covx(j,j);
    end
end
% Response Covariance Matrix (COV) Approximation
covy=zeros(length(muy));
for i=[1:length(muy)]
    for j=[1:length(muy)]
        for m=[1:length(mux)]
            for n=[1:length(mux)]
                dcov(m,n)=dydx(i,m)*dydx(j,n)*covx(m,n);
            end
        end
        covy(i,j)=sum(sum(dcov));
    end
end
% Compute Correlation (CORR) matrix calculation using COV
D=sqrt(diag(diag(covy)));   % Find the square root of the diagonal of COVY
corry=D^-1*covy*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Construct Joint Normal Probability Space with TSFD results (see equation ??)
syms y1 y2
sigy1=covy(1,1)^0.5
sigy2=covy(2,2)^0.5
p=corry(1,2);
fy1y2=((2*pi*sigy1*sigy2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((y1-muy(1))/sigy1)^2)-2*p*...
((y1-muy(1))/sigy1)*((y2-muy(2))/sigy2)+((y2-muy(2))/sigy2)^2));
% Visualize joint probability space
s=3;
ezsurf(fy1y2,[muy(1)-s*sigy1 muy(1)+s*sigy1 muy(2)-s*sigy2 muy(2)+s*sigy2],85)
%
% Compute failure probability
s=9;
```

```
pfuinv=1-dblquad(@finvsol,muy(1)-s*sigy1,maxy1,muy(2)-s*sigy2,maxy2,1e-6);
% pf=0.01562339578409 (independent input)
% pf=0.03497183732497 (dependent input rho=0.5)
```

# *B.3 Linear System with Non-normal Input*

## B.3.1 EDF Method

edflogn.m

```
% Thesis Proposal Study
% Pedagogical Example (April 13, 2003)
%
% ------------------
% Independent input
% ------------------
N=1E6;
mu=20;
sig=2;
maxy1=46.5;
maxy2=70;
mul=mean(log(random('norm',mu,sig,N,1)));
sigl=std(log(random('norm',mu,sig,N,1)));
x1=random('logn',mul,sigl,N,1);
x2=random('logn',mul,sigl,N,1);
% Visualization of input distribution
figure
[rho,xv1,xv2]=density(x1,x2,-100,-100,[14 26 14 26]);
[xv1,xv2]=meshgrid(xv1,xv2);
mesh(xv1,xv2,rho)
axis([14 26 14 26 0 0.05])
xlabel('x1')
ylabel('x2')
zlabel('density')
figure
[rho,xv1,xv2]=density(x1,x2,-30,-30,[14 26 14 26]);
contour(xv1,xv2,rho)
axis([14 26 14 26])
xlabel('x1')
ylabel('x2')
%
y1=x1+x2;
y2=x1+2*x2;
y=[y1 y2];
clear x1 x2
% Statistics
rhoy1y2=corrcoef(y)  % rho=0.9488
muV=[mean(y1) mean(y2)];    % muV=[39.99772962495611 59.99770173384864]
sigV=[std(y1) std(y2)];     % sigV=[2.87424194893095   4.54350358822563]
%
% Visualization of output distribution
figure
[rho,yv1,yv2]=density(y1,y2,-100,-100,[30 50 45 75]);
[yv1,yv2]=meshgrid(yv1,yv2);
mesh(yv1,yv2,rho)
axis([30 50 45 75 0 0.05])
xlabel('y1')
ylabel('y2')
zlabel('density')
figure
[rho,yv1,yv2]=density(y1,y2,-60,-60,[30 50 45 75]);
contour(yv1,yv2,rho,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
%
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
y=[y1 y2];
clear y1 y2
C=[0 0];
for j=[1 2];
    for i=[1:N];
        if j==1;
            limit=maxy1;
        else
            limit=maxy2;
        end
        if y(i,j)>=limit
        C(j)=C(j)+1;
        end
    end
end
Pfis=1-(1-C(1)/N)*(1-C(2)/N)        % Series event (ind) probability formula
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:N];
        if y(i,1)>=maxy1 | y(i,2)>=maxy2
```

```
        C=C+1;
        end
end
%clear y
Pfid=C/N        % Joint Probability
%
% Statistical Characterization (Minitab Dist ID feature used to determine that
%       y1 and y2 are lognormal so find distribution parameters using MATLAB
[y1muhat,y1sigmahat]=normfit(log(y1))   % y1muhat=3.6864569 and y1sigmahat=0.0717664
[y2muhat,y2sigmahat]=normfit(log(y2))   % y2muhat=4.09161953689049 and y2sigmahat=0.07560491699936
```

## B.3.2 IPDF Method

$IPDF_case2.m$

```
% ------------------------------------------------------------------------
% Thesis Proposal Study
% Pedagogical Example (April 13, 2003)
% INVERSE SOLUTION OF PROB 2
mu=20;
sig=2;
maxy1=46.5;
maxy2=70;
% Lognormal case
%mul=mean(log(random('norm',mu,sig,N,1)));
%sigl=std(log(random('norm',mu,sig,N,1)));
syms x1 x2 y1 y2
mul=mu;
sigl=sig;
% Lognormal case
fx1x2=(1/((2*pi)^0.5*sigl*x1))*exp(-0.5*((log(x1)-mul)^2)/sigl^2)*(1/((2*pi)^0.5*sigl*x2))...
*exp(-0.5*((log(x2)-mul)^2)/sigl^2)
%fx1x2=(1/((2*pi)^0.5*sigl))*exp(-0.5*((x1-mul)^2)/sigl^2)*(1/((2*pi)^0.5*sigl))...
*exp(-0.5*((x2-mul)^2)/sigl^2)
fy1y2=subs(subs(fx1x2,x1,2*y1-y2),x2,-y1+y2)
%
ezcontour(real(fy1y2),[30 50 45 75],52)
ys1=linspace(30,50,100);
ys2=linspace(45,65,100);
for i=[min(ys1):max(ys2)];
    for j=[min(ys2):max(ys2)];
        Z(i,j)=subs(subs(fy1y2,y1,ys1(i)),y2,ys2(j));
    end
end
[yy1,yy2]=meshgrid(ys1,ys2);
ezmesh(yy1,yy2,Z);
%
% Solve for P(y1>=maxy1 or y2>=maxy2)
fy1y2=simple(simplify(fy1y2));
%pfuinv=1-numeric(int(int(fy1y2,y1,0,1),y2,8,10))   % joint solution (see function call below)

%  Function call solution (hand solved) (implemented in finvsollogn.m as function for
%       integration)
function pdfinvsol=finvsol(yv1,yv2,m,s)
xx1=2*yv1-yv2;
xx2=yv2-yv1;
m=2.99058351015016;
s=0.10130405768226;
if xx1<0
    xx1=0.00001;
elseif xx1<0.0001
    xx1=0.00001;
end
if xx2<0
    xx2=0.00001;
elseif xx2<0.0001
    xx2=0.00001;
end
coeff=(1./(2*pi*s*s*(xx1).*(xx2)));
exponent=(-1/2)*(((log(xx1)-m).^2/s^2)+((log(xx2)-m).^2/s^2));
pdfinvsol=coeff.*exp(exponent);
% execute this command to perform numerical integration
pfuinv=1-dblquad(@finvsollogn,0,maxy1,0,maxy2);
% Input characteristics of responses (from EDF solution)
Efy1=40.0018;      % Solve for expected value of y1
Efy2=60.0041;      % Solve for expected value of y2
stdy1=2.8703; % Solve for std. dev. of y1
stdy2=4.5395; % Solve for std. dev. of y2
corry1y2=0.9488;    % Calcuated using EDF results  (see edflogn.m)
%corry1y2=numeric(int(int(((y1-Efy1)/stdy1)*((y2-Efy2)/stdy2)*fy1y2,y1,-inf,inf),y2,-inf,inf));
%corry1y2=dbquad(((y1-Efy1)/stdy1)*((y2-Efy2)/stdy2)*fy1y2,y1,-inf,inf),y2,-inf,inf));
%
% BPDF Method
r=8;    % Number of standard deviations
y1max=numeric(Efy1+r*stdy1);  % Calculate ymax using r*stdy1
y1min=numeric(Efy1-r*stdy1);  % Calculate ymin using r*stdy1
y2max=numeric(Efy2+r*stdy2);  % Calculate ymax using r*stdy2
y2min=numeric(Efy2-r*stdy2);  % Calculate ymin using r*stdy2
y1med=numeric((y1max+y1min)/2); % Calculate y1 median
y2med=numeric((y2max+y2min)/2); % Calculate y2 median
```

```
y1range=numeric((y1max-y1min)/2); % Calculate y1 range
y2range=numeric((y2max-y2min)/2); % Calculate y2 range
g=1+corry1y2*((y1-y1med)/y1range)*((y2-y2med)/y2range)
% Plot g function
ezmesh(g,[y1min y1max y2min y2max],50)
% Plot g/C function
C=int(int(fy1*fy2*g,y1,-inf,inf),y2,-inf,inf);
gC=g/C;
ezmesh(gC,[y1min y1max y2min y2max],50)
ezmesh(ginvsol,[y1min y1max y2min y2max],500)
% Solve for failure probability using BPDF with IPDF providing input
fy1y2SPDF=(1/C)*fy1*fy2*g
fy1y2SPDF=simple(simplify(fy1y2SPDF));
pfuinv=1-numeric(int(int(fy1y2SPDF,y1,y1min,maxy1),y2,y2min,maxy2))
% Construct g/C using INV method
ginvsolC=fy1y2/(fy1*fy2);       % solve for g/C ratio using inverse transformation solution
C=int(int(fy1*fy2*ginvsolC,y1,-inf,inf),y2,-inf,inf);
ezmesh(ginvsolC,[y1min y1max y2min y2max],500)
%pfuinv=1-numeric(int(int(fy1y2,y1,y1min,maxy1),y2,y2min,maxy2))
% Solve for failure probability using IPDF
pfuinv=1-numeric(int(int(fy1y2,y1,y1min,maxy1),y2,y2min,maxy2))
%pfuinv=1-numeric(int(int(fy1y2,y1,y1min,maxy1),y2,y2min,maxy2))
```

### B.3.3 NPDF Method

*$NPDF_case2.m$*

```
% JMW Thesis Chapter: Joint Probability Models
%
% NPDF_case1.m is the implementation of various transformation approaches for
% comparison to IPDF and BPDF
%
% ------------------------------------------------------------
% Sub case 1:  Find JPDF of responses and proceed with NPDF
% ------------------------------------------------------------
maxy1=46.5;
maxy2=70;
%
% TPDF Method 1: Nataf Method (y1 y2 are lognormal by MINITABL dist ID using A-D statistic)
syms z1 z2 y1 y2
% Use response mean and variance vectors as well as correlation coefficient to find R'(rho12')
%   Must use a numerical method function call to find rho12p given these inputs
lognmuV=[3.68645685968811 4.09161953689049];       % Vector of response means (From edflogn.m)
lognsigV=[0.07176636621301 0.07560491699936];     % Vector of response std. deviations (edflogn.m)
% Construct lognormal distributions for y1 y2
fy1=(1/((2*pi)^0.5*lognsigV(1)*y1))*exp(-0.5*((log(y1)-lognmuV(1))^2)/lognsigV(1)^2)
fy2=(1/((2*pi)^0.5*lognsigV(2)*y2))*exp(-0.5*((log(y2)-lognmuV(2))^2)/lognsigV(2)^2)
% Create joint standard normal dividend which introduces joint randomness in NATAF model
Fy1=int(fy1,y1)+0.5;    % y1 marginal cumulative dist
Fy2=int(fy2,y2)+0.5;     % y2 marginal cumulative dist
Finvy1=finverse(Fy1);    % Find inverse function of y1 CDF
Finvy2=finverse(Fy2);    % Find inverse function of y2 CDF
%
%-----------------------------------------------------------------------------------------
% Solve Case 1 (Dependent Input) using NATAF model (Numerical Integration)
% ------------------------------------------------------------
% Compute Efy1
muV=[39.99772962495611 59.99770173384864]
sigV=[2.87424194893095   4.54350358822563]
rho=0.9488;     % Determined through corrcoef of edflogn.m
cov=sigV./muV;
rhop=rho*((log(1+rho*cov(1)*cov(2)))/(rho*(log(1+cov(1)^2)*log(1+cov(2)^2))^0.5))
s=4;
pfunataf=1-dblquad(@fnataf,muV(1)-s*sigV(1),maxy1,muV(2)-s*sigV(2),maxy2,1e-6,[],fy1,fy2,...
Fy1,Fy2,rho);
%
%-----------------------------------------------------------------------------------------
% Solve Case 1 (Independent Input) using NATAF model (Numerical Integration)
% ------------------------------------------------------------
rho=0.9487;     % Determined through iterative numerical integration
          % NOTE: rhop from NATAF and rho from original space are equivalent!
s=7;
pfunataf=dblquad(@fnataf,Efy1-s*stdy1,maxy1,Efy2-s*stdy2,maxy2,1e-6,[],fy1,fy2,Fy1,Fy2,rho);
%
%  Visualization of NPDF Solution
rho=0.9843;
yy1=linspace(30,50,50);
yy2=linspace(45,75,50);
[yy1,yy2]=meshgrid(yy1,yy2);
    fy1a=5734161139222659/72057594037927936*exp(-1/16.*yy1.^2+5.*yy1-100)*pi^(1/2);
    fy2a=5734161139222659/360287970189639680*exp(-1/40.*yy2.^2-90+3.*yy2)*5^(1/2)*2^(1/2)...
    *pi^(1/2);
    Fy1a=5734161139222659/36028797018963968*pi*erf(1/4*yy1-10)+1/2;
    Fy2a=5734161139222659/360287970189639680*5^(1/2)*2^(1/2)*pi*10^(1/2)*erf(1/20*10^(1/2)...
    .*yy2-3*10^(1/2))+1/2;
    z1=norminv(Fy1a);
    z2=norminv(Fy2a);
    phi=((2*pi*(1-rho^2)^0.5)^-1)*exp(((2*rho^2-2)^-1)*(z1.^2-2*rho.*z1.*z2+z2.^2))./...
```

```
    ((normpdf(z1).*normpdf(z2)));
    fy1y2N=phi*fy1a*fy2a;
mesh(yy1,yy2,fy1y2N)
%mesh(yy1,yy2,phi)
clear yy1 yy2
yy1=linspace(30,50,50);
yy2=linspace(45,75,50);
for i=[1:length(yy1)]
    for j=[1:length(yy2)]
    fy1a=5734161139222659/72057594037927936*exp(-1/16.*yy1(i).^2+5.*yy1(i)-100)*pi^(1/2);
    fy2a=5734161139222659/360287970189639680*exp(-1/40.*yy2(j).^2-90+3.*yy2(j))*5^(1/2)*...
    2^(1/2)*pi^(1/2);
    Fy1a=5734161139222659/36028797018963968*pi*erf(1/4*yy1(i)-10)+1/2;
    Fy2a=5734161139222659/360287970189639680*5^(1/2)*2^(1/2)*pi*10^(1/2)*erf(1/20*10^(1/2)...
    .*yy2(j)-3*10^(1/2))+1/2;
    z1=norminv(Fy1a);
    z2=norminv(Fy2a);
    phi=((2*pi*(1-rho^2)^0.5)^-1)*exp(((2*rho^2-2)^-1)*(z1.^2-2*rho.*z1.*z2+z2.^2))...
    ./((normpdf(z1).*normpdf(z2)));
    fy1y2N(i,j)=phi*fy1a*fy2a;
    end
end
mesh(fy1y2N)
contour(yy1,yy2,fy1y2N,10)
axis([30 50 45 75])
xlabel('y1')
ylabel('y2')
% Verify joint randomness via rho check


% ------------------------------------------------------------------------------------
% Validation case (Der Kiureghian 1986 example 2)
% ------------------------------------------------------------
%
muy1=1;
muy2=1;
stdy1=1;
stdy2=1;
rho=0.3;
fy1=exp(-y1);
fy2=exp(-y2);
rhoo=dblquad(@drhoijval,-1E8,1E8,-1E8,1E8,1e-6,[],stdy1,stdy2,muy1,muy2,rho)
```

### B.3.4 MFOSM Method

$MFOSM_case1_2.m$

```
% JMW Thesis   8/26/03
%
% Taylor Series Finite Difference Joint Probability Method (TPDF)
% for case 2:  Linear system of equations with dependent normal input
mu=20;
sig=2;
p=0.5;
maxy1=46.5;
maxy2=70;
mux=[mu mu];    % Define input vector of variable mean values (1xn)
muy=linearfcn(mux);  % Calculate mean of responses using linearfcn function
covx=[sig*sig p*sig*sig; p*sig*sig sig*sig];     % Define input covariance vector (n by n)
% Find input correlation coefficient matrix
D=sqrt(diag(diag(covx)));   % Find the square root of the diagonal of corrx
corrx=D^-1*covx*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Sensitivity Analysis (see equation 71)
ny=2;           % Provide number of responses
for i=[1:ny]
    for j=[1:length(mux)]
        xp=mux;
        xp(j)=xp(j)+covx(j,j);
        xm=mux;
        xm(j)=xm(j)-covx(j,j);
        yp=linearfcn(xp);
        ym=linearfcn(xm);
        dydx(i,j)=(yp(i)-ym(i))/(2*covx(j,j));
%        dydx(i,j)=(muy(i)-ym(i))/covx(j,j);
    end
end
% Response Covariance Matrix (COV) Approximation
covy=zeros(length(muy));
for i=[1:length(muy)]
    for j=[1:length(muy)]
        for m=[1:length(mux)]
            for n=[1:length(mux)]
                dcov(m,n)=dydx(i,m)*dydx(j,n)*covx(m,n);
            end
        end
        covy(i,j)=sum(sum(dcov));
    end
end
% Compute Correlation (CORR) matrix calculation using COV
```

```
D=sqrt(diag(diag(covy)));   % Find the square root of the diagonal of COVY
corry=D^-1*covy*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Construct Joint Normal Probability Space with TSFD results
syms y1 y2
sigy1=covy(1,1)^0.5
sigy2=covy(2,2)^0.5
p=corry(1,2);
fy1y2=((2*pi*sigy1*sigy2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((y1-muy(1))/sigy1)^2)-2*p*...
((y1-muy(1))/sigy1)*((y2-muy(2))/sigy2)+((y2-muy(2))/sigy2)^2));
% Visualize joint probability space
s=3;
ezsurf(fy1y2,[muy(1)-s*sigy1 muy(1)+s*sigy1 muy(2)-s*sigy2 muy(2)+s*sigy2],85)
%
% Compute failure probability
s=7;
pfuinv=1-dblquad(@finvsol,muy(1)-s*sigy1,maxy1,muy(2)-s*sigy2,maxy2,1e-6);
% pf=0.03497 (dependent input)
%%%
%
%
%
% Taylor Series Finite Difference Joint Probability Method (TPDF)
% for case 2:  Linear system of equations with independent lognormal input
mu=20;
sig=2;
p=0;
maxy1=46.5;
maxy2=70;
mux=[mu mu];    % Define input vector of variable mean values (1xn)
muy=linearfcn(mux);  % Calculate mean of responses using linearfcn function
covx=[sig*sig p*sig*sig; p*sig*sig sig*sig];     % Define input covariance vector (n by n)
% Find input correlation coefficient matrix
D=sqrt(diag(diag(covx)));   % Find the square root of the diagonal of corrx
corrx=D^-1*covx*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Sensitivity Analysis (see equation 71)
ny=2;           % Provide number of responses
for i=[1:ny]
    for j=[1:length(mux)]
        xp=mux;
        xp(j)=xp(j)+covx(j,j);
        xm=mux;
        xm(j)=xm(j)-covx(j,j);
        yp=linearfcn(xp);
        ym=linearfcn(xm);
        dydx(i,j)=(yp(i)-ym(i))/(2*covx(j,j));
%        dydx(i,j)=(muy(i)-ym(i))/covx(j,j);
    end
end
% Response Covariance Matrix (COV) Approximation
covy=zeros(length(muy));
for i=[1:length(muy)]
    for j=[1:length(muy)]
        for m=[1:length(mux)]
            for n=[1:length(mux)]
                dcov(m,n)=dydx(i,m)*dydx(j,n)*covx(m,n);
            end
        end
        covy(i,j)=sum(sum(dcov));
    end
end
% Compute Correlation (CORR) matrix calculation using COV
D=sqrt(diag(diag(covy)));   % Find the square root of the diagonal of COVY
corry=D^-1*covy*D^-1;       % Find correlation matrix using equation 3.35 of Rencher (p. 69)
%
% Construct Joint Normal Probability Space with TSFD results (see equation ??)
syms y1 y2
sigy1=covy(1,1)^0.5
sigy2=covy(2,2)^0.5
p=corry(1,2);
fy1y2=((2*pi*sigy1*sigy2*(1-p^2)^0.5)^-1)*exp(((2*p^2-2)^-1)*((((y1-muy(1))/sigy1)^2)-2*p*...
((y1-muy(1))/sigy1)*((y2-muy(2))/sigy2)+((y2-muy(2))/sigy2)^2));
% Visualize joint probability space
s=3;
ezsurf(fy1y2,[muy(1)-s*sigy1 muy(1)+s*sigy1 muy(2)-s*sigy2 muy(2)+s*sigy2],85)
%
% Compute failure probability
s=9;
pfuinv=1-dblquad(@finvsol,muy(1)-s*sigy1,maxy1,muy(2)-s*sigy2,maxy2,1e-6);
% pf=0.01562339578409 (independent input)
% pf=0.03497183732497 (dependent input rho=0.5)
```

# APPENDIX C

# TURBINE BLADE PROGRAMS

## *C.1 ABLE Environment*

### C.1.1 ANSYS FEA Programs

#### *C.1.1.1 Thermo-Mechanical Material Property File: Nimonic80A*

Nimonic80A.mat

```
/COMM, ************************************************
/COMM, *****              NIMONIC 80A              *******
/COMM, *****  Compiled by Jon Wallace 9-9-2003  *******
/COMM  ***(From Jon Wallace Ph.D Thesis Fall 2003)*****
/COMM, ************************************************
MAT,1
/COMM  *****
/COMM, ***** EX/ EY/ EZ Elastic Modulus  ***
/COMM  *****
MPTEMP
MPTEMP,  1,  0.6800E+02,  2.1200E+02,  3.9200E+02,  5.7200E+02
MPTEMP,  5,  7.5200E+02,  9.3200E+02,  1.1120E+03,  1.2920E+03
MPTEMP,  9,  1.4720E+03,  1.6520E+03,  1.8320E+03,
MPDATA, EX,   1, 1,  3.3E+07,  3.2E+07,  3.1E+07,  3.0E+07
MPDATA, EX,   1, 5,  2.9E+07,  2.8E+07,  2.7E+07,  2.6E+07
MPDATA, EX,   1, 9,  2.4E+07,  2.3E+07,  2.0E+07,
/COMM  *****
/COMM  ***** POISSON RATIO
/COMM  *****
MPTEMP,  1,  2.0000E+02,  3.0000E+02,  4.0000E+02,  5.0000E+02
MPTEMP,  5,  6.0000E+02,  7.0000E+02,  8.0000E+02,  9.0000E+02
MPTEMP,  9,  1.0000E+03,  1.1000E+03,  1.2000E+03,  1.3000E+03
MPTEMP, 13,  1.4000E+03,  1.5000E+03,  1.6000E+03,  1.7000E+03
MPTEMP, 17,  1.8000E+03
MPDATA, NUXY,   1, 1,  .3000,    .3000,    .3000,    .3000
MPDATA, NUXY,   1, 5,  .3000,    .3000,    .3000,    .3000
MPDATA, NUXY,   1, 9,  .3000,    .3000,    .3000,    .3000
MPDATA, NUXY,   1, 13, .3000,    .3000,    .3000,    .3000
MPDATA, NUXY,   1, 17, .3000
MPTEMP
MPTEMP,1,   68,  212,  392,  572
MPTEMP,5,   752, 932, 1112, 1292
MPTEMP,9,  1472, 1652, 1832
MPDATA,KXX,1,1, 1.50e-4, 1.55e-4, 1.93e-4, 2.15e-4
MPDATA,KXX,1,5, 2.38e-4, 2.60e-4, 2.78e-4, 2.98e-4
MPDATA,KXX,1,9, 3.28e-4, 3.55e-4, 3.80e-4
/COMM  *****
/COMM  ***** Thermal Expansion @ 0 Deg to grain axis
/COMM  *****
MPDATA,ALPX,1,1, 7.1e-6, 7.1e-6, 7.4e-6, 7.6e-6,
MPDATA,ALPX,1,5, 7.8e-6, 8.0e-6, 8.3e-6, 8.6e-6,
MPDATA,ALPX,1,9, 9.0e-6, 9.5e-6, 10.1e-6,
/COMM  *****
/COMM  ***** Specific Heat
/COMM  *****
MPDATA,C,1,1, 41.3, 43.3, 45.4, 47.9
MPDATA,C,1,5, 50.6, 52.9, 55.3, 58.0
MPDATA,C,1,9, 60.3, 62.6, 64.9,
/COMM  *****
/COMM  ***** Density ( .296 Lb/in**3 )
```

```
/COMM  *****
MPTEMP
MPTEMP,1,70,2000
MPDATA,DENS,1,1,.76666E-3,.76666E-3
```

### *C.1.1.2 Failure Material Property File: Nimonic80A*

n80Afprop.mac

```
/comm  Nimonic 80A failure material properties
*dim,fprop,table,5,14,1
*set,fprop(0,1,1),68
*set,fprop(0,2,1),212
*set,fprop(0,3,1),392
*set,fprop(0,4,1),572
*set,fprop(0,5,1),752
*set,fprop(0,6,1),932
*set,fprop(0,7,1),1112
*set,fprop(0,8,1),1292
*set,fprop(0,9,1),1472
*set,fprop(0,10,1),1562
*set,fprop(0,11,1),1652
*set,fprop(0,12,1),1742
*set,fprop(0,13,1),1832
*set,fprop(0,14,1),2012
*set,fprop(1,0,1),1
*set,fprop(2,0,1),2
*set,fprop(3,0,1),3
*set,fprop(4,0,1),4
*set,fprop(5,0,1),5
*set,fprop(1,1,1),112
*set,fprop(1,2,1),100
*set,fprop(1,3,1),97
*set,fprop(1,4,1),97
*set,fprop(1,5,1),97
*set,fprop(1,6,1),97
*set,fprop(1,7,1),96
*set,fprop(1,8,1),95
*set,fprop(1,9,1),63
*set,fprop(1,10,1),50
*set,fprop(1,11,1),37
*set,fprop(1,12,1),23
*set,fprop(1,13,1),8
*set,fprop(1,14,1),5
*set,fprop(2,1,1),180
*set,fprop(2,2,1),170
*set,fprop(2,3,1),166
*set,fprop(2,4,1),166
*set,fprop(2,5,1),166
*set,fprop(2,6,1),166
*set,fprop(2,7,1),160
*set,fprop(2,8,1),128
*set,fprop(2,9,1),73
*set,fprop(2,10,1),55
*set,fprop(2,11,1),38
*set,fprop(2,12,1),25
*set,fprop(2,13,1),13
*set,fprop(2,14,1),10
*set,fprop(3,1,1),327.049
*set,fprop(3,2,1),327.049
*set,fprop(3,3,1),327.049
*set,fprop(3,4,1),327.049
*set,fprop(3,5,1),327.049
*set,fprop(3,6,1),327.049
*set,fprop(3,7,1),299.049
*set,fprop(3,8,1),375.151
*set,fprop(3,9,1),345.848
*set,fprop(3,10,1),383.906
*set,fprop(3,11,1),383.906
*set,fprop(3,12,1),383.906
*set,fprop(3,13,1),383.906
*set,fprop(3,14,1),383.906
*set,fprop(4,1,1),-0.117
*set,fprop(4,2,1),-0.117
*set,fprop(4,3,1),-0.117
*set,fprop(4,4,1),-0.117
*set,fprop(4,5,1),-0.117
*set,fprop(4,6,1),-0.117
*set,fprop(4,7,1),-0.1715
*set,fprop(4,8,1),-0.1727
*set,fprop(4,9,1),-0.1886
*set,fprop(4,10,1),-0.1886
*set,fprop(4,11,1),-0.1886
*set,fprop(4,12,1),-0.1886
```

```
*set,fprop(4,13,1),-0.1886
*set,fprop(4,14,1),-0.1886
*set,fprop(5,1,1),8.534E-4
*set,fprop(5,2,1),8.534E-4
*set,fprop(5,3,1),8.534E-4
*set,fprop(5,4,1),8.534E-4
*set,fprop(5,5,1),8.534E-4
*set,fprop(5,6,1),8.534E-4
*set,fprop(5,7,1),8.534E-4
*set,fprop(5,8,1),8.534E-4
*set,fprop(5,9,1),8.534E-4
*set,fprop(5,10,1),8.534E-4
*set,fprop(5,11,1),1.849E-3
*set,fprop(5,12,1),2.600E-3
*set,fprop(5,13,1),3.366E-3
*set,fprop(5,14,1),4.000E-3
*dim,fcreep,table,11,15,1
*set,fcreep(0,1,1),752
*set,fcreep(0,2,1),842
*set,fcreep(0,3,1),932
*set,fcreep(0,4,1),1022
*set,fcreep(0,5,1),1112
*set,fcreep(0,6,1),1202
*set,fcreep(0,7,1),1292
*set,fcreep(0,8,1),1382
*set,fcreep(0,9,1),1472
*set,fcreep(0,10,1),1562
*set,fcreep(0,11,1),1652
*set,fcreep(0,12,1),1742
*set,fcreep(0,13,1),1832
*set,fcreep(0,14,1),1922
*set,fcreep(0,15,1),2012
*set,fcreep(1,0,1),2
*set,fcreep(2,0,1),5
*set,fcreep(3,0,1),10
*set,fcreep(4,0,1),18
*set,fcreep(5,0,1),29
*set,fcreep(6,0,1),42
*set,fcreep(7,0,1),58
*set,fcreep(8,0,1),74
*set,fcreep(9,0,1),95
*set,fcreep(10,0,1),120
*set,fcreep(11,0,1),135
*set,fcreep(1,1,1),1000000000
*set,fcreep(1,2,1),1000000000
*set,fcreep(1,3,1),1000000000
*set,fcreep(1,4,1),1000000000
*set,fcreep(1,5,1),1000000000
*set,fcreep(1,6,1),1000000000
*set,fcreep(1,7,1),100000000
*set,fcreep(1,8,1),10000000
*set,fcreep(1,9,1),1000000
*set,fcreep(1,10,1),100000
*set,fcreep(1,11,1),10000
*set,fcreep(1,12,1),1000
*set,fcreep(1,13,1),100
*set,fcreep(1,14,1),10.00
*set,fcreep(1,15,1),4.00
*set,fcreep(2,1,1),1000000000
*set,fcreep(2,2,1),1000000000
*set,fcreep(2,3,1),1000000000
*set,fcreep(2,4,1),1000000000
*set,fcreep(2,5,1),1000000000
*set,fcreep(2,6,1),100000000
*set,fcreep(2,7,1),10000000
*set,fcreep(2,8,1),1000000
*set,fcreep(2,9,1),100000
*set,fcreep(2,10,1),10000
*set,fcreep(2,11,1),1000
*set,fcreep(2,12,1),100
*set,fcreep(2,13,1),12
*set,fcreep(2,14,1),4.00
*set,fcreep(2,15,1),0.08
*set,fcreep(3,1,1),1000000000
*set,fcreep(3,2,1),1000000000
*set,fcreep(3,3,1),1000000000
*set,fcreep(3,4,1),1000000000
*set,fcreep(3,5,1),100000000
*set,fcreep(3,6,1),10000000
*set,fcreep(3,7,1),1000000
*set,fcreep(3,8,1),500000
*set,fcreep(3,9,1),10000
*set,fcreep(3,10,1),1000
*set,fcreep(3,11,1),110
*set,fcreep(3,12,1),12
*set,fcreep(3,13,1),4
*set,fcreep(3,14,1),0.06
*set,fcreep(3,15,1),0.01
```

```
*set,fcreep(4,1,1),1000000000
*set,fcreep(4,2,1),1000000000
*set,fcreep(4,3,1),1000000000
*set,fcreep(4,4,1),100000000
*set,fcreep(4,5,1),10000000
*set,fcreep(4,6,1),1000000
*set,fcreep(4,7,1),500000
*set,fcreep(4,8,1),40000
*set,fcreep(4,9,1),3000
*set,fcreep(4,10,1),200
*set,fcreep(4,11,1),30
*set,fcreep(4,12,1),5
*set,fcreep(4,13,1),0.080
*set,fcreep(4,14,1),0.010
*set,fcreep(4,15,1),0.001
*set,fcreep(5,1,1),1000000000
*set,fcreep(5,2,1),1000000000
*set,fcreep(5,3,1),100000000
*set,fcreep(5,4,1),10000000
*set,fcreep(5,5,1),1000000
*set,fcreep(5,6,1),600000
*set,fcreep(5,7,1),80000
*set,fcreep(5,8,1),5000
*set,fcreep(5,9,1),500
*set,fcreep(5,10,1),50
*set,fcreep(5,11,1),5
*set,fcreep(5,12,1),0.080
*set,fcreep(5,13,1),0.010
*set,fcreep(5,14,1),0.010
*set,fcreep(5,15,1),0.001
*set,fcreep(6,1,1),1000000000
*set,fcreep(6,2,1),100000000
*set,fcreep(6,3,1),10000000
*set,fcreep(6,4,1),1000000
*set,fcreep(6,5,1),500000
*set,fcreep(6,6,1),100000
*set,fcreep(6,7,1),6000
*set,fcreep(6,8,1),400
*set,fcreep(6,9,1),50
*set,fcreep(6,10,1),5
*set,fcreep(6,11,1),0.08000
*set,fcreep(6,12,1),0.01000
*set,fcreep(6,13,1),0.00100
*set,fcreep(6,14,1),0.00010
*set,fcreep(6,15,1),0.00001
*set,fcreep(7,1,1),100000000
*set,fcreep(7,2,1),10000000
*set,fcreep(7,3,1),1000000
*set,fcreep(7,4,1),500000
*set,fcreep(7,5,1),100000
*set,fcreep(7,6,1),9000
*set,fcreep(7,7,1),800
*set,fcreep(7,8,1),50
*set,fcreep(7,9,1),5
*set,fcreep(7,10,1),0.08000
*set,fcreep(7,11,1),0.0100000000
*set,fcreep(7,12,1),0.0010000000
*set,fcreep(7,13,1),0.0001000000
*set,fcreep(7,14,1),0.0000100000
*set,fcreep(7,15,1),0.0000010000
*set,fcreep(8,1,1),10000000
*set,fcreep(8,2,1),1000000
*set,fcreep(8,3,1),500000
*set,fcreep(8,4,1),200000
*set,fcreep(8,5,1),10000
*set,fcreep(8,6,1),1000
*set,fcreep(8,7,1),60
*set,fcreep(8,8,1),7
*set,fcreep(8,9,1),0.08000
*set,fcreep(8,10,1),0.01000
*set,fcreep(8,11,1),0.0010000000
*set,fcreep(8,12,1),0.0001000000
*set,fcreep(8,13,1),0.0000100000
*set,fcreep(8,14,1),0.0000010000
*set,fcreep(8,15,1),0.0000001000
*set,fcreep(9,1,1),1000000
*set,fcreep(9,2,1),1000000
*set,fcreep(9,3,1),300000
*set,fcreep(9,4,1),30000
*set,fcreep(9,5,1),1000
*set,fcreep(9,6,1),70
*set,fcreep(9,7,1),6
*set,fcreep(9,8,1),0.08000
*set,fcreep(9,9,1),0.01000
*set,fcreep(9,10,1),0.00100
*set,fcreep(9,11,1),0.0001000000
*set,fcreep(9,12,1),0.0000100000
*set,fcreep(9,13,1),0.0000010000
```

```
*set,fcreep(9,14,1),0.0000001000
*set,fcreep(9,15,1),0.0000000100
*set,fcreep(10,1,1),1000000
*set,fcreep(10,2,1),600000
*set,fcreep(10,3,1),40000
*set,fcreep(10,4,1),2000
*set,fcreep(10,5,1),80
*set,fcreep(10,6,1),7
*set,fcreep(10,7,1),0.08000
*set,fcreep(10,8,1),0.01000
*set,fcreep(10,9,1),0.00100
*set,fcreep(10,10,1),0.00010
*set,fcreep(10,11,1),0.0000100000
*set,fcreep(10,12,1),0.0000010000
*set,fcreep(10,13,1),0.0000001000
*set,fcreep(10,14,1),0.0000000100
*set,fcreep(10,15,1),0.0000000010
*set,fcreep(11,1,1),1000000
*set,fcreep(11,2,1),100000
*set,fcreep(11,3,1),1000
*set,fcreep(11,4,1),100
*set,fcreep(11,5,1),8
*set,fcreep(11,6,1),0.08000
*set,fcreep(11,7,1),0.01000
*set,fcreep(11,8,1),0.00100
*set,fcreep(11,9,1),0.00010
*set,fcreep(11,10,1),0.00001
*set,fcreep(11,11,1),0.0000010000
*set,fcreep(11,12,1),0.0000001000
*set,fcreep(11,13,1),0.0000000100
*set,fcreep(11,14,1),0.0000000010
*set,fcreep(11,15,1),0.0000000001
```

### *C.1.1.3 Thermo-Mechanical Analysis FEA Macro*

run.mac

```
/batch
/comm
/comm  UPDATE DATABASE- READ BOUNDARY CONDITION FILE
/comm
/file,cfm56hot
resume
/FILNAME, Fname, 0
/prep7
/nopr
lsclear,all
nall
eall
mpdele,all,all
mpread,nimonic80A,mat
/input,n80Afprop,mac,,,1
tref,70
wsort,all
kbc,1
etchg,STT
csys,0
/nerr,5,10000000
/title,CFM56 S2B Baseline ThermoMechanical Failure  Solution (Steady State)
fini
/comm
/comm  SOLVE THERMAL SOLUTION
/comm
/input,ansys_load_ext,dat
/input,ansys_load_int,dat
/solu
antype,static
cnvtol,temp,1000,.001
eqslv,pcg
/stat,solu
solve
finish
/post1
set,last
cmsel,s,extsurf
etable,surftemp,temp
allsel,all
/comm Select either blade surface nodes/elements and process temp result
/comm Continue with Mechanical Analysis
/prep7
etchg,TTS
csys,0
lsclear,all
/input,cfm56,dispbc
/input,pressure_load,dat
fini
/comm
/comm SOLVE STRESS SOLUTION
/comm
/solu
```

```
antype,static
/stat,solu
pi=3.14159265359
n=9474*2*pi/60
ldread,temp,1,,,,cfm56,rth
omega,n
eqslv,pcg
solve
/post1
set,last
/comm FAILURE ANALYSES
/comm
/comm Read in stress values into etable
cmsel,s,extsurf
etable,surfseqv,s,eqv
smult,surfseqv,surfseqv,,0.001,
/comm Create failure response etables (zero valued initially)
sadd,gustress,surfseqv,,0,0,
sadd,nfat,surfseqv,,0,0,
sadd,tcreep,surfseqv,,0,0,
/comm Random Variable Material Property Adders
*set,dsy,0
*set,dsu,0
*set,dsfp,0
*set,dbexp,0
*set,dcreep,0
*set,screep,0.329
*set,oxilimit,70
/comm Execute failure analysis macro
allsel,all
/input,failure,mac
/comm
*dim,output,array,1,12
*dim,output2,array,1,15
*set,output(1,1),Tcrep1
*set,output(1,2),Screp1
*set,output(1,3),crep1max
*set,output(1,4),Tcrep2
*set,output(1,5),Screp2
*set,output(1,6),crep2max
*set,output(1,7),Tcrep3
*set,output(1,8),Screp3
*set,output(1,9),crep3max
*set,output(1,10),Tcrep4
*set,output(1,11),Screp4
*set,output(1,12),crep4max
*set,output2(1,1),Tfat3
*set,output2(1,2),Sfat3
*set,output2(1,3),fat3max
*set,output2(1,4),Tfat4
*set,output2(1,5),Sfat4
*set,output2(1,6),fat4max
*set,output2(1,7),Tgu3
*set,output2(1,8),Sgu3
*set,output2(1,9),gu3max
*set,output2(1,10),Tgu4
*set,output2(1,11),Sgu4
*set,output2(1,12),gu4max
*set,output2(1,13),gumax
*set,output2(1,14),fatmax
*set,output2(1,15),crepmax
fini
*CREATE,ansuitmp
*CFOPEN,'Fname','outputa',' ','append'
*VWRITE,output(1,1),output(1,2),output(1,3),output(1,4),output(1,5),output(1,6),output(1,7),
output(1,8),output(1,9),output(1,10),output(1,11),output(1,12)
(1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',
1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5)
*CFCLOS
*END
/INPUT,ansuitmp
!*
*CREATE,ansuitmp
*CFOPEN,'Fname','outputb',' ','append'
*VWRITE,output2(1,1),output2(1,2),output2(1,3),output2(1,4),output2(1,5),output2(1,6),output2(1,7),
output2(1,8),output2(1,9),output2(1,10),output2(1,11),output2(1,12),output2(1,13),output2(1,14),
output2(1,15)
(1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',
1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5)
*CFCLOS
*END
/INPUT,ansuitmp
!*
*CREATE,ansuitmp
*CFOPEN,'Fname','results',' ','append'
*VWRITE,results(1,1),results(1,2),results(1,3),results(1,4),results(1,5),results(1,6)
(1PE12.5,' ',1PE12.5' ',1PE12.5' ',1PE12.5' ',1PE12.5,' ',1PE12.5)
*CFCLOS
*END
/INPUT,ansuitmp
!*
/exit
```

### *C.1.1.4 Failure Analysis FEA Macro*

failure.mac

```
*dim,results,array,6939,6,1
*Do,I,1,6939
        *set,element,ext_e(I,1)
        *get,stress,etab,2,elem,element
        *get,temper,etab,1,elem,element
        *set,syield,fprop(1,temper,1)+dsy
        *set,sult,fprop(2,temper,1)+dsu
        *set,sfp,fprop(3,temper,1)+dsfp
        *set,bexp,fprop(4,temper,1)+dbexp
        *set,creep,exp((1/(A+B*log(stress)))+(QR/(temper+459.67))+dcreep*screep)
        *set,fat,0.5*(0.5*stress/(sfp*(1-0.5*stress/sult)))**(1/bexp)
        detab,element,nfat,fat
        detab,element,tcreep,creep
        detab,element,gustress,stress/sult
          *set,results(I,1),element
          *set,results(I,2),temper
          *set,results(I,3),stress
          *set,results(I,4),fat
          *set,results(I,5),creep
          *set,results(I,6),stress/sult
*ENDDO
allsel,all
/comm Failure Analysis Postprocessing
/comm
/comm CREEP1
cmsel,s,crephot1
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tcrep1,etab,1,elem,enumb
*get,Screp1,etab,2,elem,enumb
*get,Crep1max,etab,5,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,Crep,etab,5,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,Crep,lt,Crep1max,then
                *set,Crep1max,Crep
                    *set,Tcrep1,T
                    *set,Screp1,S
        *endif
*ENDDO
/comm CREEP2
cmsel,s,crephot2
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tcrep2,etab,1,elem,enumb
*get,Screp2,etab,2,elem,enumb
*get,Crep2max,etab,5,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,Crep,etab,5,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,Crep,lt,Crep2max,then
                *set,Crep2max,Crep
                    *set,Tcrep2,T
                    *set,Screp2,S
        *endif
*ENDDO
/comm CREEP3
cmsel,s,crephot3
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tcrep3,etab,1,elem,enumb
*get,Screp3,etab,2,elem,enumb
*get,Crep3max,etab,5,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,Crep,etab,5,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,Crep,lt,Crep3max,then
                *set,Crep3max,Crep
                    *set,Tcrep3,T
                    *set,Screp3,S
        *endif
*ENDDO
/comm CREEP4
cmsel,s,crephot4
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tcrep4,etab,1,elem,enumb
*get,Screp4,etab,2,elem,enumb
*get,Crep4max,etab,5,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,Crep,etab,5,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,Crep,lt,Crep4max,then
                *set,Crep4max,Crep
                    *set,Tcrep4,T
```

```
                    *set,Screp4,S
        *endif
*ENDDO
/comm FAT3
cmsel,s,fathot3
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tfat3,etab,1,elem,enumb
*get,Sfat3,etab,2,elem,enumb
*get,fat3max,etab,4,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,fat,etab,4,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,fat,lt,fat3max,then
                *set,fat3max,fat
                    *set,Tfat3,T
                    *set,Sfat3,S
        *endif
*ENDDO
/comm FAT4
cmsel,s,fathot4
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tfat4,etab,1,elem,enumb
*get,Sfat4,etab,2,elem,enumb
*get,fat4max,etab,4,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,fat,etab,4,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,fat,lt,fat4max,then
                *set,fat4max,fat
                    *set,Tfat4,T
                    *set,Sfat4,S
        *endif
*ENDDO
/comm GU3
cmsel,s,guhot3
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tgu3,etab,1,elem,enumb
*get,Sgu3,etab,2,elem,enumb
*get,gu3max,etab,3,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,gu,etab,3,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,gu,lt,gu3max,then
                *set,gu3max,gu
                    *set,Tgu3,T
                    *set,Sgu3,S
        *endif
*ENDDO
/comm GU3
cmsel,s,guhot4
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tgu4,etab,1,elem,enumb
*get,Sgu4,etab,2,elem,enumb
*get,gu4max,etab,3,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,gu,etab,3,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,gu,lt,gu4max,then
                *set,gu4max,gu
                    *set,Tgu4,T
                    *set,Sgu4,S
        *endif
*ENDDO
allsel,all
/comm gumax
cmsel,s,extsurf
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tgu,etab,1,elem,enumb
*get,Sgu,etab,2,elem,enumb
*get,gumin,etab,3,elem,enumb
*get,gumax,etab,3,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,gu,etab,3,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,gu,lt,gumax,then
                *set,gumax,gu
                    *set,Tgu,T
                    *set,Sgu,S
        *endif
        *if,gu,lt,gumin,then
                *set,gumin,gu
        *endif
*ENDDO
```

```
/comm fatmax
cmsel,s,extsurf
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tfat,etab,1,elem,enumb
*get,Sfat,etab,2,elem,enumb
*get,fatmin,etab,4,elem,enumb
*get,fatmax,etab,4,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,fat,etab,4,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,fat,lt,fatmax,then
                *set,fatmax,fat
                    *set,Tfat,T
                    *set,Sfat,S
        *endif
        *if,fat,lt,fatmin,then
                *set,fatmin,fat
        *endif
*ENDDO
/comm crepmax
cmsel,s,extsurf
*get,ecount,elem,,count
*get,enumb,elem,,num,min
*get,Tcrep,etab,1,elem,enumb
*get,Screp,etab,2,elem,enumb
*get,crepmin,etab,5,elem,enumb
*get,crepmax,etab,5,elem,enumb
*Do,I,1,ecount-1
        *get,enumb,elem,enumb,nxth
        *get,crep,etab,5,elem,enumb
          *get,T,etab,1,elem,enumb
          *get,S,etab,2,elem,enumb
        *if,crep,lt,crepmax,then
                *set,crepmax,crep
                    *set,Tcrep,T
                    *set,Screp,S
        *endif
        *if,crep,gt,crepmin,then
                *set,crepmin,crep
        *endif
*ENDDO
allsel,all
```

## C.1.2 PERL Programs

### *C.1.2.1 DOE Table Variable Value Converter*

convert.pl

```
#!/usr/bin/perl

#  DOE.PL  created 6/01/01 by Jon Wallace
#
#  Synopsis:
#    CONVERT.PL is a script created to to automate the process of reading
#    in and processing DOE cases from doe_table.txt.  The script reads in
#    the DOE case list, converts all normalized values to absolute values
#    using a reference file (bvalues.txt) containing the baseline values
#    for each variable and finally creates a new file called
#    doe_table_abs.txt with the real variable values included
#
#  Update 1/02/02:   Added variables for YFT ANSYS cooling hole convergence/solution
#           removed NU, and replaced creep constant variables with
#           one creep constant variation parameter (see DOE description)
#  Update 11/18/01:  Added 8 hole diameter scaling feature   JMW
#  Update 11/26/01:  Added bulk creep calculation feature    JMW
#  Modified 9/11/03: MOdified for thesis implementation  JMW
#
# Initialize Variables
#
$x= (); #array position
$i= (); #counter
$j= (); #counter
$line_baseline_file = ""; # String
$line_properties= "";  # String containing all new property values
$EXL= ();  # Lower Elastic Modulus Room Temp Value
$KXXL= (); # Lower Thermal Conductivity Room Temp Value
$ALPXL= (); # Lower thermal expansion coefficient room temp value
$DENSL= (); # Lower density room temp value
$CNL= (); # Orr-sherby-dorn lower creep constant variation value
$DSYL= (); # Yield stress variation term
$DSUL= (); # Ultimate stress variation term
$DSFPL= (); # Fatigue strength coefficient variation term
$DBL= ();  # Fatigue exponent variation term
$RNL= (); # Lower rotational speed value
$SINTTL= (); # Internal temperature field scalar
$SEXTTL= (); # External gas temperature field scalar
$EXH= ();  # Lower Elastic Modulus Room Temp Value
```

```
$KXXH= (); # Lower Thermal Conductivity Room Temp Value
$ALPXH= (); # Lower thermal expansion coefficient room temp value
$DENSH= (); # Lower density room temp value
$CNH= (); # Orr-sherby-dorn lower creep constant variation value
$DSYH= (); # Yield stress variation term
$DSUH= (); # Ultimate stress variation term
$DSFPH= (); # Fatigue strength coefficient variation term
$DBH= ();  # Fatigue exponent variation term
$RNH= (); # Lower rotational speed value
$SINTTH= (); # Internal temperature field scalar
$SEXTTH= (); # External gas temperature field scalar
$values = (); # list containing sequence of material prop. values
@values = ();  # Array containing sequence of matl. prop. values
$data1 = ();  # Dummy variable
$data2 = ();  # Dummy variable
$data3 = ();  # Dummy variable
$data4 = ();  # Dummy variable
$data5 = ();  # Dummy variable
$data6 = ();  # Dummy variable
$data7 = ();  # Dummy variable
$data8 = ();  # Dummy variable
$data9 = ();  # Dummy variable

# Read in baseline variable values
open(IN_PROPERTIES,"bvalues.txt") or die "Can't open baseline variable value file: $!\n";
while ($line_properties=<IN_PROPERTIES>)
    {
    if($line_properties=~ /EX=/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($EXL,$EXH)=split(/,/, $data2);
    }
    if($line_properties=~ /KXX=/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($KXXL,$KXXH)=split(/,/, $data2);
    }
    if($line_properties=~ /ALPX/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($ALPXL,$ALPXH)=split(/,/, $data2);
    }
    if($line_properties=~ /DENS/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($DENSL,$DENSH)=split(/,/, $data2);
    }
    if($line_properties=~ /CN/)
        {
                ($data1,$data2)=split(/=/, $line_properties,2);
                ($CNL,$CNH)=split(/,/, $data2);
        }
    if($line_properties=~ /RN/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($RNL,$RNH)=split(/,/, $data2);
    }
    if($line_properties=~ /DSY/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($DSYL,$DSYH)=split(/,/, $data2);
    }
    if($line_properties=~ /DSU/)
        {
                ($data1,$data2)=split(/=/, $line_properties,2);
                ($DSUL,$DSUH)=split(/,/, $data2);
        }
    if($line_properties=~ /DSFP/)
        {
                ($data1,$data2)=split(/=/, $line_properties,2);
                ($DSFPL,$DSFPH)=split(/,/, $data2);
        }
        if($line_properties=~ /DB/)
        {
                ($data1,$data2)=split(/=/, $line_properties,2);
                ($DBL,$DBH)=split(/,/, $data2);
        }
    if($line_properties=~ /SINTT/)
        {
                ($data1,$data2)=split(/=/, $line_properties,2);
                ($SINTTL,$SINTTH)=split(/,/, $data2);
        }
    if($line_properties=~ /SEXTT/)
    {
        ($data1,$data2)=split(/=/, $line_properties,2);
        ($SEXTTL,$SEXTTH)=split(/,/, $data2);
    }
}
@LOW=($EXL,$KXXL,$ALPXL,$DENSL,$CNL,$DSYL,$DSUL,$DSFPL,$DBL,$RNL,$SINTTL,$SEXTTL);
```

```
@HIGH=($EXH,$KXXH,$ALPXH,$DENSH,$CNH,$DSYH,$DSUH,$DSFPH,$DBH,$RNH,$SINTTH,$SEXTTH);
chomp @LOW;
chomp @HIGH;
close IN_PROPERTIES;

open(IN_BASELINE_FILE,"doe_table.txt") or die "Can't open DOE Table: $!\n";
open(OUT_DOE,">doe_table_abs.txt") or die "Can't create new DOE TABLE file: $!\n";

use Time::localtime;
$now = ctime();
print OUT_DOE "# $now\n";
print OUT_DOE "# NUMBER, EX, KXX, ALPX, DENS, CN, SY, SU, SFP, B, RN, INTT, EXTT\n";

$j=1;   # Counter for case number assignment

while($line_baseline_file=<IN_BASELINE_FILE>)
    {
    if($line_baseline_file =~ /,/)
    {
        ($data1,$data2,$data3,$data4,$data5,$data6,$data7,$data8,$data9,$data10,$data11,
        $data12,$data13,$data1)=split(/,/, $line_baseline_file);
        @values = ($data2,$data3,$data4,$data5,$data6,$data7,$data8,$data9,$data10,
        $data11,$data12,$data13);
        $i=0;
        foreach $x (@values)
        {
            $x = ( (  $x + 1  ) / 2 ) * ( $HIGH[$i] - $LOW[$i] ) + $LOW[$i];
            $i++;
        }
        @values=("CASE=$j",@values);
        $values = join(', ', @values);
        $line_baseline_file = $values;
        print OUT_DOE "$line_baseline_file\n";
        $j++;   # Increment case number assignment
    }
    else {}
}
close IN_BASELINE_FILE;
close OUT_DOE;
print "Completed converting DOE case list\n";
```

### *C.1.2.2 Material Property File Creator*

mchange.pl

```
#!/usr/bin/perl

#  MCHANGE.PL  created 6/01/01 by Jon Wallace
#
#  Synopsis:
#    MCHANGE.PL is a script created to to automate the process of updating
#    the ANSYS material property input file.  This script is intended for
#    material property input files where the property varies as a function
#    of temperature.  The script obtains the new material property from
#    mvalues.txt and computes a scalar based on the dividend of the new value
#    and the baseline value provided in a baseline input file, matl.txt.
#
#  Update 1/02/02:  Removed NU variable   JMW
#  Modified 9/11/03: MOdified for thesis implementation JMW
#
# Initialize Variables
#
$line_properties= "";  # String containing all new property values
$CASE= (); # Case Number
$EX= ();  # Elastic Modulus Room Temp Value
$ALPX= (); # Thermal expansion coefficient
$DSY= (); # Yield strength (ksi)
$DSU= (); # Ultimate strength (ksi)
$DSFP= (); # Elastic fatigue constant
$DB= (); # Elastic fatigue exponent
$KXX= (); # Thermal Conductivity Room Temp Value
$DENS= (); # density room temp value
$CN= (); # Orr-sherby-dorn creep constant variation parameter
$SEXTT= (); # New external convective gas temperature field boundary condition scalar value
$SINTT= (); # New internal convective gas temperature field bc scalar value
$RN= (); # Rotor speed value
$data= "";  # Dummy Data Variable for Splitting
$BEX= ();   # Elastic Modulus Array Scalar
$BKXX= ();   # Thermal conductivity Array Scalar
$BALPX= ();   # Thermal expansion coefficient Array scalar
$values = (); # list containing sequence of material prop. values
@values= ();  # Array containing sequence of matl. prop. values
@data1 = "";  # Dummy variable
@data2 = "";  # Dummy variable
@data3 = "";  # Dummy variable
@data4 = "";  # Dummy variable
$now = (); # Time variable
# Read in new room temperature values (and case number)

open(IN_PROPERTIES,"mvalues.txt") or die "Can't open new matl prop value file: $!\n";

while ($line_properties=<IN_PROPERTIES>)
```

```
        {
        if($line_properties=~ /EX=/)
        {
                ($data,$EXN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /CASE/)
        {
                ($data,$CASE)=split(/=/, $line_properties);
                chomp $CASE;
        }
        if($line_properties=~ /KXX/)
        {
                ($data,$KXXN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /ALPX/)
        {
                ($data,$ALPXN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /DENS/)
        {
                ($data,$DENSN)=split(/=/, $line_properties);
        }
        else {}
}
close IN_PROPERTIES;

# Read in baseline room temperature values to determine proportionality scalar
open(IN_BASELINE_FILE,"matl.mat") or die "Can't open baseline material input file: $!\n";

while($line_baseline_file=<IN_BASELINE_FILE>)
        {
        if($line_baseline_file =~ /MPDATA\s*,\s*EX,\s*1,\s*1,\s*\d*/)
        {
        ($data,$data,$data,$data,$EX)=split(/,/, $line_baseline_file);
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*KXX,\s*1,\s*1,\s*\d*/)
        {
        ($data,$data,$data,$data,$KXX)=split(/,/, $line_baseline_file);
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*ALPX,\s*1,\s*1,\s*\d*/)
        {
        ($data,$data,$data,$data,$ALPX)=split(/,/, $line_baseline_file);
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*DENS,\s*1,\s*1,\s*\d*/)
        {
        ($data,$data,$data,$data,$DENS)=split(/,/, $line_baseline_file);
        }
        else {}
}

# Compute scalar for each material property
$BEX = $EXN / $EX;
$BKXX = $KXXN / $KXX;
$BALPX = $ALPXN / $ALPX;
$BDENS = $DENSN / $DENS;

close IN_BASELINE_FILE;

# Update All Material Property Values based on proportionality scalars and
# write to a new file called newmatl.txt



open(IN_BASELINE_FILE,"matl.mat") or die "Can't open baseline material input file: $!\n";
@OUTPUT=('matl','_',$CASE,'.mat');
$OUTPUT=join('',@OUTPUT);
open(OUT_MATL,">$OUTPUT") or die "Can't create newmatl file: $!\n";

print OUT_MATL "/COMM Case $CASE\n";
use Time::localtime;
$now = ctime();
print OUT_MATL "/COMM $now\n";

while($line_baseline_file=<IN_BASELINE_FILE>)
        {
        if($line_baseline_file =~ /MPDATA\s*,\s*EX/)
        {
                ($data1,$data2,$data3,$data4,$values)=split(/,/, $line_baseline_file, 5);
                @values = split(/,/, $values);

                foreach $x (@values) {
                        $x = $x * $BEX;
                }
                $values = join(', ', @values);
                @newline = ($data1,$data2,$data3,$data4,$values);
                $line_baseline_file = join(', ', @newline);
                print OUT_MATL "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*KXX/)
        {
                ($data1,$data2,$data3,$data4,$values)=split(/,/, $line_baseline_file, 5);
                @values = split(/,/, $values);

                foreach $x (@values) {
```

```
                        $x = $x * $BKXX;
                }
                $values = join(', ', @values);
                @newline = ($data1,$data2,$data3,$data4,$values);
                $line_baseline_file = join(', ', @newline);
                print OUT_MATL "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*ALPX/)
        {
                ($data1,$data2,$data3,$data4,$values)=split(/,/, $line_baseline_file, 5);
                @values = split(/,/, $values);

                foreach $x (@values) {
                        $x = $x * $BALPX;
                }
                $values = join(', ', @values);
                @newline = ($data1,$data2,$data3,$data4,$values);
                $line_baseline_file = join(', ', @newline);
                print OUT_MATL "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /MPDATA\s*,\s*DENS/)
        {
                ($data1,$data2,$data3,$data4,$values)=split(/,/, $line_baseline_file, 5);
                @values = split(/,/, $values);

                foreach $x (@values) {
                        $x = $x * $BDENS;
                }
                $values = join(', ', @values);
                @newline = ($data1,$data2,$data3,$data4,$values);
                $line_baseline_file = join(', ', @newline);
                print OUT_MATL "$line_baseline_file\n";
        }
        else {
                print OUT_MATL "$line_baseline_file";
        }
}
close IN_BASELINE_FILE;
close OUT_MATL;
print "Completed Creating Case $CASE Matl. File\n";
```

### *C.1.2.3 FEA Boundary Condition File Creator*

bchange.pl

```
#!/usr/bin/perl

#  BCHANGE.PL  created 6/01/01 by Jon Wallace
#
#  Synopsis:
#    BCCHANGE.PL is a script created to to automate the process of updating
#    the ANSYS external and internal boundary conditions including the
#    pressure boundary condition file.  This script uses
#    a scalar defined in the DOE case for both the external and internal
#    boundary conditions (2 variables) to update both the heat transfer film
#    coefficients and the gas bulk temperature at that surface.  In addition
#    the script requires the current rotor speed specified in DOE case for a
#    scalar on the film coefficient.  A simple uniform scalar is also applied to
#    the pressures defined in the pressure_load baseline BC file.
#
#  Update 12/13/01:  Altered to only scale h and T values in ansys_load_ext.dat
#                    file independtly and create new ext.dat for case $
#
# Initialize Variables
#
$line_properties= "";  # String containing all new property values
$CASE= (); # Case Number
$RN= ();   # New Rotational Speed Value (rad/sec)
$SEXTT= ();   # New External Gas Temperature Field Scalar
$SINTT= ();   # New internal gas temperature field scalar
$data= "";   # Dummy Data Variable for Splitting
$data1= "";   # Dummy Data Variable for Splitting
$data2= "";   # Dummy Data Variable for Splitting
$data3= "";   # Dummy Data Variable for Splitting
$data4= "";   # Dummy Data Variable for Splitting
$data5= "";   # Dummy Data Variable for Splitting
$values= (); # list containing sequence of material prop. values
@values= ();  # Array containing sequence of matl. prop. values
$now = (); # Time variable

# Read in new rotor speed and boundary condition scalars (and case number)

open(IN_VALUES,"mvalues.txt") or die "Can't open mvalues.txt file: $!\n";

while ($line_properties=<IN_VALUES>)
        {
        if($line_properties=~ /CASE/)
```

```
        {
                ($data,$CASE)=split(/=/, $line_properties);
                chomp $CASE;
        }
        if($line_properties=~ /RN/)
        {
                ($data,$RN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /SEXTT/)
        {
                ($data,$SEXTT)=split(/=/, $line_properties);
        }
            if($line_properties=~ /SINTT/)
        {
                ($data,$SINTT)=split(/=/, $line_properties);
        }
}
close IN_VALUES;

# Convert RN (rad/sec) to RN (rpm)

#print "$RN\n";
$RN = $RN * 60 / (2 * 3.14159265359);
#print "$RN\n";

# Update All External Boundary Condition Values (h and Ts) and create new EXT and INT BC FILE
# Add case number to the new EXT INT BC files

open(IN_BASELINE_FILE,"ansys_load_ext.dat") or die "Can't open baseline external bc file: $!\n";
@OUTPUT=('ansys_load_ext','_',$CASE,'.dat');
$OUTPUT=join('',@OUTPUT);
open(OUT_EXT,">$OUTPUT") or die "Can't create new external BC file: $!\n";

print OUT_EXT "/COMM Case $CASE\n";
use Time::localtime;
$now = ctime();
print OUT_EXT "/COMM $now\n";

while($line_baseline_file=<IN_BASELINE_FILE>)
        {
        if($line_baseline_file =~ /TVAL=/)
        {
                ($data,$TVAL)=split(/=/, $line_baseline_file);
                $TVAL= $SEXTT*($TVAL + 460) - 460;  # Add temperature instead of scale
                @newline = ($data,$TVAL);
                $line_baseline_file = join('= ', @newline);
                print OUT_EXT "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /SET\s*,\s*CFAC/)
        {
                ($data1,$data2,$data3)=split(/,/,$line_baseline_file,3);
                $data3= "144*$RN";
                @newline = ($data1,$data2,$data3);
                $line_baseline_file = join(', ', @newline);
                print OUT_EXT "$line_baseline_file\n";
        }
        else {
                print OUT_EXT "$line_baseline_file";
        }
}
close IN_BASELINE_FILE;
close OUT_EXT;

open(IN_BASELINE_FILE,"ansys_load_int.dat") or die "Can't open baseline external
boundary condition file: $!\n";
@OUTPUT=('ansys_load_int','_',$CASE,'.dat');
$OUTPUT=join('',@OUTPUT);
open(OUT_INT,">$OUTPUT") or die "Can't create new internal BC file: $!\n";

print OUT_INT "/COMM Case $CASE\n";
use Time::localtime;
$now = ctime();
print OUT_INT "/COMM $now\n";

while($line_baseline_file=<IN_BASELINE_FILE>)
        {
        if($line_baseline_file =~ /TVAL=/)
        {
                ($data,$TVAL)=split(/=/, $line_baseline_file);
                $TVAL= $SINTT*($TVAL + 460) - 460;  # Add temperature instead of scalar
                @newline = ($data,$TVAL);
                $line_baseline_file = join('= ', @newline);
                print OUT_INT "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /SET\s*,\s*CFAC/)
        {
                ($data1,$data2,$data3)=split(/,/,$line_baseline_file,3);
                $data3= "144*$RN";
                @newline = ($data1,$data2,$data3);
                $line_baseline_file = join(', ', @newline);
                print OUT_INT "$line_baseline_file\n";
        }
        else {
```

```
                print OUT_INT "$line_baseline_file";
        }
}
close IN_BASELINE_FILE;
close OUT_INT;
print "Completed Creating Case $CASE EXT and INT BC Files\n";
```

### *C.1.2.4 ANSYS Run File Creator*

rchange.pl

```
#!/usr/bin/perl

#  RCHANGE.PL  created 6/06/01 by Jon Wallace
#
#  Synopsis:
#    RCHANGE.PL is a script created to automate the process of updating
#    the ANSYS batch file.  Currently, only the rotational speed (turbine
#    rotor speed) and the cooling hole diameters (8) are altered by this
#    program.  But, other parameters can easily be added should the need arise.
#
#  Update 11/18/01:  Added 8 hole diameter scaling feature   JMW
#  Update 11/26/01:  Added creep constant input into macro creation
#           Note:  creep constants aren't added into matl_case#.mat
#                  file but rather into the ANSYS macro file.
#  Update 12/17/01:  Changed script to create 3 run files instead of two for
#                making the program more modular
#                    run_h.mac for the cooling hole diameter change
#                run_t.mac for the thermal analysis run
#                run_m.mac for the mechanical analysis run
#  Modified 9/11/03:  Modified to implement JMW thesis    JMW
#
# Initialize Variables
#
$line_properties= "";  # String containing all new property values
$CASE= (); # Case Number
$RN= ();  # New Rotational Speed Value (rad/sec)
$CN= (); # Orr-sherby-dorn creep A constant
$DSY= (); # Cooling Hole 1 scalar
$DSU= (); # Cooling Holes 2-5 scalar
$DB= (); # Cooling Holes 6-8 scalar
$values = (); # list containing sequence of prop. values
@values= ();  # Array containing sequence of prop. values
@data = "";  # Dummy variable
use Time::localtime; # Initiate system time stamp
$now = (); # Local Time Stamp Variable

# Read in the new rotor speed and cooling hole diameter values and case number

open(IN_PROPERTIES,"mvalues.txt") or die "Can't open mvalues.txt file: $!\n";

while ($line_properties=<IN_PROPERTIES>)
        {
        if($line_properties=~ /RN/)
        {
                ($data,$RN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /CN/)
        {
                ($data,$CN)=split(/=/, $line_properties);
        }
        if($line_properties=~ /DSY/)
        {
                ($data,$DSY)=split(/=/, $line_properties);
        }
    if($line_properties=~ /DSU/)
        {
                ($data,$DSU)=split(/=/, $line_properties);
        }
    if($line_properties=~ /DSFP/)
        {
                ($data,$DSFP)=split(/=/, $line_properties);
        }
        if($line_properties=~ /DB/)
        {
                ($data,$DB)=split(/=/, $line_properties);
        }
    if($line_properties=~ /CASE/)
        {
                ($data,$CASE)=split(/=/, $line_properties);
                chomp $CASE;
        }
}
close IN_PROPERTIES;

#
open(IN_BASELINE_FILE,"run.mac") or die "Can't open baseline analysis ANSYS macro file: $!\n";
@OUTPUT=('run',$CASE,".mac");
$OUTPUT=join('',@OUTPUT);
#print "$OUTPUT\n";
open(OUT_MATL,">$OUTPUT") or die "Can't create case run_t file: $!\n";
```

```
while($line_baseline_file=<IN_BASELINE_FILE>)
        {
        if($line_baseline_file =~ /FILNAME/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,"case_$CASE",$data3);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /CFOPEN,'Fname'/)
        {
                ($data1,$data2,$data3,$data4,$data5)=split(/,/, $line_baseline_file);
                @newline = ($data1,"'case_$CASE'",$data3,$data4,$data5);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /VREAD/)
        {
        ($data1,$data2,$data9,$values,$data3,$data4,$data5,$data6,$data7,$data8)=split(/,/,
        $line_baseline_file);
        @newline = ($data1,$data2,$data9,"case_$CASE",$data3,$data4,$data5,$data6,$data7,$data8);
        $line_baseline_file = join(',', @newline);
        print OUT_MATL "$line_baseline_file\n";
        }
        elsif($line_baseline_file =~ /mpread/)
        {
                ($data1,$values,$data2)=split(/,/, $line_baseline_file);
                @newline = ($data1,"matl_$CASE",$data2);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /input,ansys_load_ext/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,"ansys_load_ext_$CASE",$data3);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /input,ansys_load_int/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,"ansys_load_int_$CASE",$data3);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /ldread/)
        {
                ($data1,$data2,$data3,$data4,$data5,$data6,$data7,$data8)=split(/,/,
                $line_baseline_file);
                $data7=$data7;
                @newline = ($data1,$data2,$data3,$data4,$data5,$data6,"case_$CASE",$data8);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /n=/)
        {
                ($data1,$values)=split(/=/, $line_baseline_file);
                @newline = ($data1,$RN);
                $line_baseline_file = join('=', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /set,dcreep/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,$data2,$CN);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /set,dsy/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,$data2,$DSY);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /set,dsu/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,$data2,$DSU);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        elsif($line_baseline_file =~ /set,dsfp/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,$data2,$DSFP);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
```

```
        elsif($line_baseline_file =~ /set,dbexp/)
        {
                ($data1,$data2,$data3)=split(/,/, $line_baseline_file);
                @newline = ($data1,$data2,$DB);
                $line_baseline_file = join(',', @newline);
                print OUT_MATL "$line_baseline_file";
        }
        else {
                print OUT_MATL "$line_baseline_file";
        }
}
print OUT_MATL "/COMM Case $CASE\n";
use Time::localtime;
$now = ctime();
print OUT_MATL "/COMM $now\n";

close IN_BASELINE_FILE;
close OUT_MATL;
print "Completed Creating Case $CASE ANSYS macro\n";
```

### *C.1.2.5 Master Blade FEA Program*

run.pl

```
#!/usr/bin/perl

#   run.pl is the master script that coordinates multiple
#   slave scripts to automatically create a new ANSYS FEA
#   case based on a list of DOE cases.
#
#   Update 1/02/02:   Added YFT to ANSYS convergence routine (uses ANSYS run_h.mac,
#           run_yin.mac, runyout.mac, run_t.mac, run_m.mac)
#   Update 11/18/01:  Added 8 hole diameter scaling feature   JMW
#   Update 11/26/01:  Added creep constant input into macro creation
#                         Note:  creep constants aren't added into matl_case#.mat
#                                file but rather into the ANSYS macro file.
#   Modification 9/5/03: Altered to accomodate and implement framework (cfm56 2nd stage blade)
#
# Initialize variables
$CASE = (); # Case number
$EX= ();  # Elastic Modulus Room Temp Value
$ALPX= (); # Thermal expansion coefficient
$DSY= (); # Yield strength (ksi)
$DSU= (); # Ultimate strength (ksi)
$DSFP= (); # Elastic fatigue constant
$DB= (); # Elastic fatigue exponent
$KXX= (); # Thermal Conductivity Room Temp Value
$DENS= (); # density room temp value
$CN= (); # Orr-sherby-dorn creep constant variation parameter
$SEXTT= (); # New external convective gas temperature field boundary condition scalar value
$SINTT= (); # New internal convective gas temperature field bc scalar value
$RN= (); # Rotor speed value
#
# Run support scripts
#system("convert.pl");
#system("create_etable.pl");

# Read in DOE Table (doe_table_abs.txt) perform tasks for each case (row)
$CASE=45;
open(IN_DOE,"doe_table_abs.txt") or die "Can't open DOE file: $!\n";
while($line_DOE_file=<IN_DOE>)
    {
    open(OUT,">mvalues.txt");
    if($line_DOE_file =~ /CASE/)
        {
        ($CASEF,$EX,$KXX,$ALPX,$DENS,$CN,$DSY,$DSU,$DSFP,$DB,$RN,$SINTT,$SEXTT)=split(/,/,
        $line_DOE_file);
        print OUT "$CASEF\n";
                print OUT "EX=$EX\n";
        print OUT "KXX=$KXX\n";
        print OUT "ALPX=$ALPX\n";
        print OUT "DENS=$DENS\n";
        print OUT "CN=$CN\n";
                print OUT "DSY=$DSY\n";
                print OUT "DSU=$DSU\n";
        print OUT "DSFP=$DSFP\n";
        print OUT "DB=$DB\n";
        print OUT "RN=$RN\n";
                print OUT "SINTT=$SINTT\n";
                print OUT "SEXTT=$SEXTT\n";
        close OUT;
        # Iterate over each DOE case to setup of input files and run ANSYS for each
        print "Starting Case $CASE\n";
        print "Creating all case $CASE static input files\n";
        system("mchange.pl");
                system("bchange.pl");
                system("rchange.pl");
        print "Runing case $CASE Thermo-Mechanical Failure Analysis\n";
            system("/home/asdl/asdl/ansys_inc/v70/ansys/bin/ansys70 -p ANSYSRF -j
```

```
            case_$CASE -s read -b nolist < run$CASE.mac > case_$CASE.out");
        system("cat case_$CASE.outputa >> allcases.outputa");
        system("cat case_$CASE.outputb >> allcases.outputb");
        system("rm case_$CASE.db");
        system("rm matl_$CASE.mat");
        system("cp mvalues.txt case$CASE.mvalues.txt");
        system("rm case_$CASE.esav");
        system("rm case_$CASE.emat");
                system("rm case_$CASE.mntr");
        system("rm case_$CASE.PCS");
                system("rm case_$CASE.PCF");
        system("rm case_$CASE.rth");
        system("rm case_$CASE.rst");
        system("rm run$CASE.mac");
                system("rm ansys_load_ext_$CASE.dat");
        system("rm ansys_load_int_$CASE.dat");
        print "Completed Case $CASE Analysis\n";
        $CASE++;
        }
        else{}
    }
close IN_DOE;
```

## C.1.3 MATLAB Programs

### *C.1.3.1 Baseline Reliability Assessment*

bladevalidation.m

```
% JMW Thesis
% CFM56 Turbofan jet engine turbine blade sensititivy and failure analysis
%       Note:  Reference ABLE3 results
%
%   This program uses the RSE for each of three hot spots created using ABLE3
%   10/3/03  Modified to use truncated distributions of parameters!
%
% ------------------------------------------------------------------------
% Implementation Validation of Blade Reliability (Monte Carlo Simulation)
% ------------------------------------------------------------------------
x=1E6; % Number of MCS simulations
%
% Input variables (8 cycle, 9 material)
mucycle=[35000 0.745 0 0.885 0.9 0.87 0.93 0.935 ];
stdcycle=[2204 0.0745 7 0.022 0.023 0.022 0.023 0.023];
covcycle=[stdcycle(1)^2 0 0 0 0 0 0 0; 0 stdcycle(2)^2 0 0 0 0 0 0; 0 0 stdcycle(3)^2 0 0 0 0 0;
          0 0 0 stdcycle(4)^2 0 0 0 0; 0 0 0 0 stdcycle(5)^2 0 0 0; 0 0 0 0 0 stdcycle(6)^2 0 0;
          0 0 0 0 0 0 stdcycle(7)^2 0; 0 0 0 0 0 0 0 stdcycle(8)^2];
% Create sample-based cycle input
Altitudel=random('norm',35000,2204,x,1);
Machl=random('norm',0.745,0.0745,x,1);
dTal=random('norm',0,7,x,1);
etafl=random('norm',0.885,0.022,x,1);
etalpcl=random('norm',0.9,0.023,x,1);
etahpcl=random('norm',0.87,0.022,x,1);
etahptl=random('norm',0.93,0.023,x,1);
etalptl=random('norm',0.935,0.023,x,1);
%
% Convert Cycle Input into Coded Values
Low=[31500  0.6705  -27 0.84075 0.855   0.8265  0.8835  0.88825];
High=[38500 0.8195  27 0.92925  0.945   0.9135  0.9765  0.98175];
Altitude=-1+2*(Altitudel-Low(1))/(High(1)-Low(1));
clear Altitudel
Mach=-1+2*(Machl-Low(2))/(High(2)-Low(2));
clear Machl
dTa=-1+2*(dTal-Low(3))/(High(3)-Low(3));
clear dTal
etaf=-1+2*(etafl-Low(4))/(High(4)-Low(4));
clear etafl
etalpc=-1+2*(etalpcl-Low(5))/(High(5)-Low(5));
clear etalpcl
etahpc=-1+2*(etahpcl-Low(6))/(High(6)-Low(6));
clear etahpcl
etahpt=-1+2*(etahptl-Low(7))/(High(7)-Low(7));
clear etahptl
etalpt=-1+2*(etalptl-Low(8))/(High(8)-Low(8));
clear etalptl
%
% Cycle Analysis (RSE)
Tc=%% not provided;
Tg=%% not provided;
N=%% not provided;
clear Altitude Mach dTa etaf etalpc etahpc etahpt etalpt
%cycle=[Tc Tg N];
%save cycle cycle -ascii
%clear cycle Tc Tg N
%
%
%
```

```
% Material Input Statistics
%  EX  KXX  ALPX DENS  CN  SY  SU  SFP  B
mumatl=[33e6 1.49898E-4 7.05555E-6 7.66665E-4 0 112 180 327.0486 -0.117];
covmatl=[(0.05*mumatl(1))^2 0 0 0 0 0 0 0 0; 0 (0.02*mumatl(2))^2 0 0 0 0 0 0 0;...
 0 0 (0.02*mumatl(3))^2 0 0 0 0 0 0;
         0 0 0 (0.02*mumatl(4))^2 0 0 0 0 0; 0 0 0 0 1 0 0 0 0; 0 0 0 0 0 ...
         (0.04*mumatl(6))^2 0 0 0; 0 0 0 0 0 0 (0.04*mumatl(7))^2 0 0;
         0 0 0 0 0 0 0 (0.019*mumatl(8))^2 0.81*(0.019*mumatl(8))*(-0.09*mumatl(9));...
          0 0 0 0 0 0 0 0.81*(-0.019*mumatl(8))*(0.09*mumatl(9))  (0.09*mumatl(9))^2];
D=sqrt(diag(diag(covmatl)));   % Find the square root of the diagonal of COVY
corrmatl=D^-1*covmatl*D^-1;
clear D
%
% Create Monte Carlo Simulation Sampling Routine for Multivariate Joint Normal Input
        % Ref: Johnson, M.E., Multivariate Statistical Simulation, 1987
Y=random('Normal',0,1,length(mumatl),x);   % Standard normal sampling matrix
%for i=[1:length(mumatl)]
%    for j=[1:x]
%    while abs(Y(i,j))>1.0
%        Y(i,j)=random('Normal',0,1,1,1);
%    end
%    end
%end
L=chol(covmatl)';                          % Find lwr triangular matrix from Cholesky Factorization
X=L*Y + mumatl'*ones(1,x);                 % Find multivariate X input vector through joint normal
                                            % sampling
clear Y L;
covX=cov(X');                             % Check cov matrix
corrX=corrcoef(X');                       % Check corr coefficient matrix with input corr matrix
%
% Define RSE input variables
EXP=X(1,:); KXXP=X(2,:); ALPXP=X(3,:); DENSP=X(4,:); CNP=X(5,:); SYP=X(6,:); SUP=X(7,:);
SFPP=X(8,:); BP=X(9,:); RNP=N;
INTTP=Tc; EXTTP=Tg;
clear X;
%
mux=mumatl;
% Convert to Coded values
Low=[31350000 6.91E-06 0.0001469 0.000751331 -1 -4.5 -7.2 -6.2 -0.01053 8126.09855 0.969 0.9737];
High=[34650000 7.20E-06 0.000152896 0.000781998 1 4.5 7.2 6.2 0.01053 10821.89775 1.031 1.0263];
%
EX=-1+2*(EXP-Low(1))/(High(1)-Low(1));
KXX=-1+2*(KXXP-Low(3))/(High(3)-Low(3));
ALPX=-1+2*(ALPXP-Low(2))/(High(2)-Low(2));
DENS=-1+2*(DENSP-Low(4))/(High(4)-Low(4));
CN=-1+2*(CNP-Low(5))/(High(5)-Low(5));
SY=-1+2*((SYP-mux(6))-Low(6))/(High(6)-Low(6));
SU=-1+2*((SUP-mux(7))-Low(7))/(High(7)-Low(7));
SFP=-1+2*((SFPP-mux(8))-Low(8))/(High(8)-Low(8));
B=-1+2*((BP-mux(9))-Low(9))/(High(9)-Low(9));
RN=-1+2*(RNP'-Low(10))/(High(10)-Low(10));
INTT=-1+2*((INTTP'/1248.7)-Low(11))/(High(11)-Low(11));
EXTT=-1+2*((EXTTP'/2401.4)-Low(12))/(High(12)-Low(12));
%
clear EXP KXXP ALPXP DENSP CNP SYP SUP SFPP BP RNP INTTP EXTTP;
%
% Specify Intermediate variable RSEs
Sc1=%% not provided;
Tc1=%% not provided;
Sc3=%% not provided;
Tc3=%% not provided;
Tf3=%% not provided;
Sf3=%% not provided;
%rsecreep1=%% not provided;
clear EX KXX ALPX DENS SY SU SFP B RN INTT EXTT;
%
% Temperature dependent material properties table
TEMP=[68 212 392 572 752 932 1112 1292 1472 1652 1832];          % Temperature range
SU=[180 170 166 166 166 166 160 128 73  38  13];                 % Ultimate tensile strength
SFP=[327.049 327.049 327.049 327.049 327.049 327.049 299.049 ...
375.151 345.848 383.906 383.906];   % Fatigue strength coefficient
SFP=0.8*SFP;
SB=[-0.117 -0.117 -0.117 -0.117 -0.117 -0.1715 -0.1727 -0.1886...
 -0.1886 -0.1886]; % Fatigue exponent
A=-0.031632;  BC=0.003;  Q=110000;   SE=0.329;    % Creep constants
        %   A=-0.031632;  BC=0.003055;  Q=106000;   SE=0.329;    % Creep constants
%
% Failure Analysis postprocessor
    % Creep Analysis
    creep1=exp((1./(A + BC.*log(Sc1)))+(Q./(Tc1 + 459.67)) + CN*SE);
    creep3=exp((1./(A + BC.*log(Sc3)))+(Q./(Tc3+459.67))+CN*SE);
    % Fatigue Analysis
    fat3=0.5*(0.5*Sf3./interp1(TEMP,SFP,Tf3)./(1-0.5*Sf3./interp1(TEMP,SU,Tf3))).^...
    (1./interp1(TEMP,SB,Tf3));
%
% Response Output Matrix
Y=[Tc1' Sc1' creep1' Tc3' Sc3' creep3' Tf3' Sf3' fat3'];
clear Sc1 Tc1 Sc3 Tc3 Sf3 Tf3 creep1 creep3 fat3 CN;
%
% Find variance and correlation coefficient matrices
```

```
covy=cov(Y);
%save RSEcovy.dat covy -ascii
corry=corrcoef(Y);
%save RSEcorry.dat corry -ascii
% ------------------------------------
% Empirical plots Generation
% ------------------------------------
%
%
%
% ------------------------------------
% Joint Probability Plotting
% ------------------------------------
figure
[rho,xv1,xv2]=density(Y(:,6),Y(:,9),-100,-100,[0 1E4 0 1E4]);
[xv1,xv2]=meshgrid(xv1,xv2);
meshc(xv1,xv2,rho)
%contour(xv1,xv2,rho,200)
axis([0 8000 0 8000])
zlabel('Probability Density')
xlabel('Fatigue (Region 3)')
ylabel('Creep (Region 3)')
% Empirical plots Points calculations
clear Failure
Failure(:,1)=sort(Y(:,3));
Failure(:,2)=sort(Y(:,6));
Failure(:,3)=sort(Y(:,9));
%
i=[1:x];
pFailure=i/x;
% Joint Prob CDF
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
k=1;
for j=[60:60:6000]
    l=1;
    while min(Failure(l,:))<j
        if Failure(l,1)<j | Failure(l,2)<j | Failure(l,3)<j
        l=l+1;
        end
    end
    Pfj(k)=l/x;
    k=k+1;
end
% Independent and Marginal Probability Calculations
k=1;
for j=[60:60:6000]
    C=[0 0 0];
    for m=[1:3]
        fail=Failure(1,m);   i=1;
        while fail<j
            i=i+1;  C(m)=C(m)+1;  fail=Failure(i,m);
        end
        Pfc(k,m)=C(m)/x;
    end
    Pfi(k)=1-(1-Pfc(k,1))*(1-Pfc(k,2))*(1-Pfc(k,3));   k=k+1;
end
life=[60:60:6000];
save jpfmfosm.dat Pf -ascii
%
%plot(Failure(:,1),pFailure,'-',Failure(:,2),pFailure,'-.',Failure(:,3),pFailure,'--')
%plot(Failure(:,1)',pFailure,'-',Failure(:,2)',pFailure,'-.',Failure(:,3)',pFailure,...
'--',life,Pfj,':',life,Pfi,'d:')
plot(life,Pfc(:,1),'-',life,Pfc(:,2),'-.',life,Pfc(:,3),'--',life,Pfj,':',life,Pfi,'d:')
xlabel('Time');
ylabel('Cumulative Probability');
axis([0 6000 0 0.03]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)','Joint Probability...
 Solution (Dependent)','Joint Probability Solution (Independent)');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)','Joint Probability...
 Solution (Dependent)','Joint Probability Solution (Independent)',0);
% Temperature response plots
%
plot(Failure(:,1),pFailure,'-',Failure(:,2),pFailure,'-.',Failure(:,3),pFailure,'--',life,Pf,':')
xlabel('Time');
ylabel('Cumulative Probability');
axis([0 6000 0 0.02]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',0);
% Temperature response plots
[N,c1]=hist(Y(:,1),1000);
[N,c3]=hist(Y(:,4),1000);
[N,f3]=hist(Y(:,7),1000);
plot(c1,N,'-',c3,N,'-.',f3,N,':')
xlabel('Temperature (Fahrenheit)');
ylabel('Frequency');
axis([0 5E10 0 2e4]);
legend('Region 1 Temperature','Region 3 Temperature');
legend('Region 1 Temperature','Region 3 Temperature',0);
figure
% Failure response plots
[N,c1]=hist(Y(:,3),500);
```

```
[N,c3]=hist(Y(:,6),500);
[N,f3]=hist(Y(:,9),500);
plot(c3,N,'-',f3,N,':')
xlabel('Failure Response (Region 3)');
ylabel('Frequency');
axis([0 7e5 0 3.5e4]);
legend('Creep','Fatigue');
legend('Creep','Fatigue',0);
%
% ------------------------------------------------
% PROBABILITY ANALYSES                            ----
% ------------------------------------------------
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Independent Input)
% ------------------------------------------------------------------------------
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 3 yrs
fatlimit=5475;    % avg of 5.0 trips/cycles per day, 365 days per year, for 3 years
gulimit=0.8;
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 3 yrs
fatlimit=5475;    % avg of 3.0 trips/cycles per day, 365 days per year, for 3 years
gulimit=0.8;
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
clear Ci Cj
Ci=[0 0 0];
Cj=0;
for j=[3 6 9];
    for i=[1:x];
        if j==1 | j==2;
            limit=creeplimit;
        else
            limit=fatlimit;
        end
        if Y(i,j)<=limit
        C(j/3)=C(j/3)+1;
        end
    end
end
Pfcs=1-(1-C(1)/x)*(1-C(2)/x)*(1-C(3)/x);        % Series event (ind) probability formula
                % Pfcs=0.0219   Pf=[0.0000    0.0141    0.0079]    10/14/03
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Dependent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 3 yrs
fatlimit=5475;    % avg of 3.0 trips/cycles per day, 365 days per year, for 3 years
gulimit=0.8;
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:x];
        if Y(i,3)<=creeplimit | Y(i,6)<=creeplimit | Y(i,9)<=fatlimit
        C=C+1;
        end
end
%clear y
Pc2c3=C/x        % Joint probability
                % Pfcd=0.0151   10/14/03
%
%
```

### *C.1.3.2 Blade Joint Probability Methods*

bladejpmodels.m

```
% JMW Thesis
% CFM56 Turbofan jet engine turbine blade sensititivy and failure analysis
%       Note:  Reference ABLE3 results
%
%
%   This program uses the RSE for each of three hot spots created using ABLE3
%
% ------------------------------------------------
% Joint Probability Models Evaluation (USING VALIDATION CYCLE
% STATISTICS!!!)
% ------------------------------------------------
%
%
n=1E6;                                  % Set number of simulations
x=n;
%
% Cycle Model Input Statistics  (see cfm56cycle.m for results)
```

```
%  Rotor Speed      Coolant temp     Gas Temp
% ---------------------
% Multiple responses
% ---------------------
% Specify sample-based information
mucycle=1e3*[ 9.58781539502712   1.25503747183137   2.41061039507026];     % sample mean
covcycle=1.0e+006*[1.71454850556157   0.03854013895860   0.05863647417329; % sample covariance
   0.03854013895860   0.00149918386416   0.00208285995048;
   0.05863647417329   0.00208285995048   0.00402965110828];
D=sqrt(diag(diag(covcycle)));   % Find the square root of the diagonal of COVY
corrcycle=D^-1*covcycle*D^-1;
% Find mu and variance for lognormal distribution (reference J.L. Devore
% Probability and Statistics)
syms v
    for i=[1:length(mucycle)]
        varcycleln(i)=numeric(solve(exp(2*(log(mucycle(i))-v/2)+v)*(exp(v)-1)-covcycle(i,i)));
    end
mucycleln=log(mucycle)-varcycleln/2;
% Compute transformed lognormal correlations and find lognormal covariance
% matrix
del=mucycleln./sqrt(varcycleln);
[l,w]=size(corrcycle);          % Get dimensions of sample-based correlation matrix
for i=[1:l]
    for j=[1:w]
    rholn(i,j)=corrcycle(i,j)*((log(1+corrcycle(i,j)*del(i)*del(j)))/(corrcycle(i,j)*...
    sqrt(log(1+del(i)^2)*log(1+del(j)^2))));
    covcycleln(i,j)=rholn(i,j)*sqrt(varcycleln(i))*sqrt(varcycleln(j));
    % Compute covariance matrix element
    %covcycleln(i,j)=corrcycle(i,j)*sqrt(varcycleln(i))*sqrt(varcycleln(j));
    end
end
%
Y=random('Normal',0,1,length(mucycleln),n);   % Standard normal sampling matrix
L=chol(covcycleln)';             % Find lower triangular matrix from Cholesky Factorization
Xcycle=exp(L*Y + mucycleln'*ones(1,n));
%
% Material Input Statistics
%  EX  KXX  ALPX DENS  CN  SY  SU  SFP  B
mumatl=[33e6 1.49898E-4 7.05555E-6 7.66665E-4 0 112 180 327.0486 -0.117];
covmatl=[(0.05*mumatl(1))^2 0 0 0 0 0 0 0 0; 0 (0.02*mumatl(2))^2 0 0 0 0 0 0 0;...
 0 0 (0.02*mumatl(3))^2 0 0 0 0 0 0;
         0 0 0 (0.02*mumatl(4))^2 0 0 0 0 0; 0 0 0 0 1 0 0 0 0; 0 0 0 0 0...
          (0.04*mumatl(6))^2 0 0 0; 0 0 0 0 0 0 (0.04*mumatl(7))^2 0 0;
         0 0 0 0 0 0 0 (0.019*mumatl(8))^2 0.81*(0.019*mumatl(8))*(-0.09*mumatl(9)); ...
         0 0 0 0 0 0 0 0.81*(-0.019*mumatl(8))*(0.09*mumatl(9))  (0.09*mumatl(9))^2];
D=sqrt(diag(diag(covmatl)));   % Find the square root of the diagonal of COVY
corrmatl=D^-1*covmatl*D^-1;
clear D
%
% Create Monte Carlo Simulation Sampling Routine for Multivariate Joint Normal Input
       % Ref: Johnson, M.E., Multivariate Statistical Simulation, 1987
Y=random('Normal',0,1,length(mumatl),n);   % Standard normal sampling matrix
L=chol(covmatl)';                % Find lower triangular matrix from Cholesky Factorization
Xmatl=L*Y + mumatl'*ones(1,n);  % Find multivariate X input vector through joint normal sampling
clear Y L;
covXmatl=real(cov(Xmatl'));                          % Check cov matrix
corrXmatl=corrcoef(Xmatl');               % Check corr coefficient matrix with input corr matrix
%
% Construct Complete Input Covariance Matrix and Mean Vector
% EX  KXX  ALPX DENS  CN  SY  SU  SFP  B 'Rotor Speed' 'Coolant temp' 'Gas Temp'
mux=[mumatl mucycle];
lm=length(covmatl);
lc=length(covcycle);
covx=[covmatl zeros(lm,lc); zeros(lc,lm) covcycle];         % Covariance matrix
corrx=sqrt(diag(diag(covx)))^-1*covx*sqrt(diag(diag(covx)))^-1; % Correlation coefficient matrix

% Define RSE input variables
EXP=Xmatl(1,:); KXXP=Xmatl(2,:); ALPXP=Xmatl(3,:); DENSP=Xmatl(4,:); CNP=Xmatl(5,:);
SYP=Xmatl(6,:); SUP=Xmatl(7,:); SFPP=Xmatl(8,:); BP=Xmatl(9,:); RNP=Xcycle(1,:);
INTTP=Xcycle(2,:); EXTTP=Xcycle(3,:);
clear Xmatl Xcycle;
%
% Convert to Coded values
Low=[31350000 6.91E-06 0.0001469 0.000751331 -1 -4.5 -7.2 -6.2 -0.01053 8126.09855 0.969 0.9737];
High=[34650000 7.20E-06 0.000152896 0.000781998 1 4.5 7.2 6.2 0.01053 10821.89775 1.031 1.0263];
%
EX=-1+2*(EXP-Low(1))/(High(1)-Low(1));
KXX=-1+2*(KXXP-Low(3))/(High(3)-Low(3));
ALPX=-1+2*(ALPXP-Low(2))/(High(2)-Low(2));
DENS=-1+2*(DENSP-Low(4))/(High(4)-Low(4));
CN=-1+2*(CNP-Low(5))/(High(5)-Low(5));
SY=-1+2*((SYP-mux(6))-Low(6))/(High(6)-Low(6));
SU=-1+2*((SUP-mux(7))-Low(7))/(High(7)-Low(7));
SFP=-1+2*((SFPP-mux(8))-Low(8))/(High(8)-Low(8));
B=-1+2*((BP-mux(9))-Low(9))/(High(9)-Low(9));
RN=-1+2*(RNP-Low(10))/(High(10)-Low(10));
INTT=-1+2*((INTTP/1248.7)-Low(11))/(High(11)-Low(11));
EXTT=-1+2*((EXTTP/2401.4)-Low(12))/(High(12)-Low(12));
%
clear EXP KXXP ALPXP DENSP CNP SYP SUP SFPP BP RNP INTTP EXTTP;
```

```
%
% Specify Intermediate variable RSEs
Sc1=%% not provided;
Tc1=%% not provided;
Sc3=%% not provided;
Tc3=%% not provided;
Tf3=%% not provided;
Sf3=%% not provided;
%rsecreep1=%% not provided;
clear EX KXX ALPX DENS SY SU SFP B RN INTT EXTT;
%
% Temperature dependent material properties table
TEMP=[68 212 392 572 752 932 1112 1292 1472 1652 1832];     % Temperature range
SU=[180 170 166 166 166 166 160 128 73  38  13];     % Ultimate tensile strength
SFP=[327.049 327.049 327.049 327.049 327.049 327.049...
 299.049 375.151 345.848 383.906 383.906];   % Fatigue strength coefficient
SFP=0.8*SFP;
SB=[-0.117 -0.117 -0.117 -0.117 -0.117 -0.1715 -0.1727...
 -0.1886 -0.1886 -0.1886];   % Fatigue exponent
A=-0.031632;  BC=0.003;  Q=110000;   SE=0.329;    % Creep constants
       %   A=-0.031632;  BC=0.003055;  Q=106000;   SE=0.329;    % Creep constants
%
% Failure Analysis postprocessor
    % Creep Analysis
    creep1=exp((1./(A + BC.*log(Sc1)))+(Q./(Tc1 + 459.67)) + CN*SE);
    creep3=exp((1./(A + BC.*log(Sc3)))+(Q./(Tc3+459.67))+CN*SE);
    % Fatigue Analysis
    fat3=0.5*(0.5*Sf3./interp1(TEMP,SFP,Tf3)./(1-0.5*Sf3./interp1(TEMP,SU,Tf3)))...
    .^(1./interp1(TEMP,SB,Tf3));
%
% Response Output Matrix
Y=[Tc1' Sc1' creep1' Tc3' Sc3' creep3' Tf3' Sf3' fat3'];
clear Sc1 Tc1 Sc3 Tc3 Sf3 Tf3 creep1 creep3 fat3 CN;
%
% Find variance and correlation coefficient matrices
covy=cov(Y);
%save RSEcovy.dat covy -ascii
corry=corrcoef(Y);
%save RSEcorry.dat corry -ascii

%
% ------------------------------------
% Joint Probability Analysis
% ------------------------------------
figure
[rho,xv1,xv2]=density(Y(:,6),Y(:,9),-100,-100,[0 1E4 0 1E4]);
[xv1,xv2]=meshgrid(xv1,xv2);
mesh(xv1,xv2,rho)
xlabel('Fatigue Life')
ylabel('Creep Rupture Life')

% ------------------------------------
% Empirical plots Generation
% ------------------------------------
%
%
% Empirical plots Points calculations
clear Failure
Failure(:,1)=sort(Y(:,3));
Failure(:,2)=sort(Y(:,6));
Failure(:,3)=sort(Y(:,9));
%
i=[1:x];
pFailure=i/x;
% Joint Prob CDF
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
k=1;
clear C
for j=[60:60:6000]
    C=0;
    for i=[1:x]
        if Failure(i,1)<j | Failure(i,2)<j | Failure(i,3)<j
        C=C+1;
        end
    end
    Pf(k)=C/x;
    k=k+1
end
life=[60:60:6000];
save jpfmfosm.dat Pf -ascii
%
plot(Failure(:,1),pFailure,'-',Failure(:,2),pFailure,'-.',Failure(:,3)...
,pFailure,'--',life,Pf,':')
%plot(Failure(:,1),pFailure,'-',Failure(:,2),pFailure,'-.',Failure(:,3),...
pFailure,'--')
xlabel('Time');
ylabel('Cumulative Probability');
axis([0 6000 0 0.02]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',...
'Joint Probability Solution');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',...
'Joint Probability Solution',0);
```

```
% Temperature response plots
[N,c1]=hist(Y(:,1),1000);
[N,c3]=hist(Y(:,4),1000);
[N,f3]=hist(Y(:,7),1000);
plot(c1,N,'-',c3,N,'-.',f3,N,':')
xlabel('Temperature (Fahrenheit)');
ylabel('Frequency');
axis([0 5E10 0 2e4]);
legend('Region 1 Temperature','Region 3 Temperature');
legend('Region 1 Temperature','Region 3 Temperature',0);
% Failure response plots
[N,c1]=hist(Y(:,3),1000);
[N,c3]=hist(Y(:,6),1000);
[N,f3]=hist(Y(:,9),1000);
plot(c1,N,'-',c3,N,'-.',f3,N,':')
xlabel('Life (hours)');
ylabel('Frequency');
axis([0 5E10 0 1e4]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',0);
%
% ------------------------------------------------
% PROBABILITY ANALYSES                             ----
% ------------------------------------------------
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Independent Input)
% ------------------------------------------------------------------------------
%
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=18250;  % avg of 5 hrs per day, 365 days per year, for 10 yrs
fatlimit=18250;    % avg of 5.0 trips/cycles per day, 365 days per year, for 10 years
gulimit=0.8;
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
clear C
C=[0 0 0];
for j=[3 6 9];
    for i=[1:x];
        if j==1 | j==2;
            limit=creeplimit;
        else
            limit=fatlimit;
        end
        if Y(i,j)<=limit
        C(j/3)=C(j/3)+1;
        end
    end
end
Pfcs=1-(1-C(1)/x)*(1-C(2)/x)*(1-C(3)/x);        % Series event (ind) probability formula
                % Pfcs=0.0219   Pf=[0.0000    0.0141    0.0079]    10/14/03
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Dependent Responses, Independent Input)-
% ------------------------------------------------------------------------------
%
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Dependent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 3 yrs
fatlimit=5475;    % avg of 3.0 trips/cycles per day, 365 days per year, for 3 years
gulimit=0.8;
%
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:x];
        if Y(i,3)<=creeplimit | Y(i,6)<=creeplimit | Y(i,9)<=fatlimit
        C=C+1;
        end
end
%clear y
Pfcd=C/x        % Joint probability
                % Pfcd=0.0151   10/14/03
%
%
```

### *C.1.3.3 Conventional Blade Reliability*

bladeconventional.m

```
% JMW Thesis
% CFM56 Turbofan jet engine turbine blade sensititivy and failure analysis
%       Note:  Reference ABLE3 results
%
%   This program uses the RSE for each of three hot spots created using ABLE3
%
% ------------------------------------------------
% Joint Probability Models Evaluation (USING VALIDATION CYCLE
% STATISTICS!!!)
% ------------------------------------------------
%
n=1E6;                                  % Set number of simulations
x=n;
%
% Cycle Model Input Statistics  (see cfm56cycle.m for results)
%  Rotor Speed      Coolant temp     Gas Temp
% --------------------
% Multiple responses
% --------------------
% Specify sample-based information
mucycle=1e3*[ 9.58781539502712   1.25503747183137   2.41061039507026];     % sample mean
covcycle=1.0e+006*[1.71454850556157   0   0; % sample covariance
   0   0.00149918386416   0;
   0   0   0.00402965110828];
D=sqrt(diag(diag(covcycle)));   % Find the square root of the diagonal of COVY
corrcycle=D^-1*covcycle*D^-1;
% Find mu and variance for lognormal distribution (reference J.L. Devore
% Probability and Statistics)
%
% Material Input Statistics
%  EX  KXX  ALPX DENS  CN  SY  SU  SFP  B
mumatl=[33e6 1.49898E-4 7.05555E-6 7.66665E-4 0 112 180 327.0486 -0.117];
covmatl=[(0.05*mumatl(1))^2 0 0 0 0 0 0 0 0; 0 (0.02*mumatl(2))^2 0 0 0 0 0 0 0;...
 0 0 (0.02*mumatl(3))^2 0 0 0 0 0 0;
         0 0 0 (0.02*mumatl(4))^2 0 0 0 0 0; 0 0 0 0 1 0 0 0 0; 0 0 0 0 0 ...
         (0.04*mumatl(6))^2 0 0 0; 0 0 0 0 0 0 (0.04*mumatl(7))^2 0 0;
         0 0 0 0 0 0 0 (0.019*mumatl(8))^2 0; 0 0 0 0 0 0 0 0  (0.09*mumatl(9))^2];
D=sqrt(diag(diag(covmatl)));   % Find the square root of the diagonal of COVY
corrmatl=D^-1*covmatl*D^-1;
clear D
%
% Construct Complete Input Covariance Matrix and Mean Vector
% EX  KXX  ALPX DENS  CN  SY  SU  SFP  B 'Rotor Speed' 'Coolant temp' 'Gas Temp'
mux=[mumatl mucycle];
lm=length(covmatl);
lc=length(covcycle);
covx=[covmatl zeros(lm,lc); zeros(lc,lm) covcycle];         % Covariance matrix
corrx=sqrt(diag(diag(covx)))^-1*covx*sqrt(diag(diag(covx)))^-1;
% Correlation coefficient matrix
%
% Create Monte Carlo Simulation Sampling Routine for Multivariate Joint Normal Input
        % Ref: Johnson, M.E., Multivariate Statistical Simulation, 1987
Y=random('Normal',0,1,length(mux),n);   % Standard normal sampling matrix
L=chol(covx)';              % Find lower triangular matrix from Cholesky Factorization
X=L*Y + mux'*ones(1,n);% Find multivariate X input vector through joint normal sampling
clear Y L;
covX=real(cov(X'));                             % Check cov matrix
corrX=corrcoef(X');              % Check corr coefficient matrix with input corr matrix
%
%
% Define RSE input variables
EXP=X(1,:); KXXP=X(2,:); ALPXP=X(3,:); DENSP=X(4,:); CNP=X(5,:); SYP=X(6,:); SUP=X(7,:);
SFPP=X(8,:); BP=X(9,:); RNP=X(10,:);
INTTP=X(11,:); EXTTP=X(12,:);
clear X;
%
% Convert to Coded values
Low=[31350000 6.91E-06 0.0001469 0.000751331 -1 -4.5 -7.2 -6.2 -0.01053 8126.09855...
 0.969 0.9737];
High=[34650000 7.20E-06 0.000152896 0.000781998 1 4.5 7.2 6.2 0.01053 10821.89775...
 1.031 1.0263];
%
EX=-1+2*(EXP-Low(1))/(High(1)-Low(1));
KXX=-1+2*(KXXP-Low(3))/(High(3)-Low(3));
ALPX=-1+2*(ALPXP-Low(2))/(High(2)-Low(2));
DENS=-1+2*(DENSP-Low(4))/(High(4)-Low(4));
CN=-1+2*(CNP-Low(5))/(High(5)-Low(5));
SY=-1+2*((SYP-mux(6))-Low(6))/(High(6)-Low(6));
SU=-1+2*((SUP-mux(7))-Low(7))/(High(7)-Low(7));
SFP=-1+2*((SFPP-mux(8))-Low(8))/(High(8)-Low(8));
B=-1+2*((BP-mux(9))-Low(9))/(High(9)-Low(9));
RN=-1+2*(RNP-Low(10))/(High(10)-Low(10));
INTT=-1+2*((INTTP/1248.7)-Low(11))/(High(11)-Low(11));
EXTT=-1+2*((EXTTP/2401.4)-Low(12))/(High(12)-Low(12));
%
clear EXP KXXP ALPXP DENSP CNP SYP SUP SFPP BP RNP INTTP EXTTP;
%
% Specify Intermediate variable RSEs
```

```
Sc1=%% not provided;
Tc1=%% not provided;
Sc3=%% not provided;
Tc3=%% not provided;
Tf3=%% not provided;
Sf3=%% not provided;
clear EX KXX ALPX DENS SY SU SFP B RN INTT EXTT;
%
% Temperature dependent material properties table
TEMP=[68 212 392 572 752 932 1112 1292 1472 1652 1832];          % Temperature range
SU=[180 170 166 166 166 166 160 128 73  38  13];         % Ultimate tensile strength
SFP=[327.049 327.049 327.049 327.049 327.049 327.049 299.049 375.151 345.848 383.906...
 383.906];   % Fatigue strength coefficient
SFP=0.8*SFP;
SB=[-0.117 -0.117 -0.117 -0.117 -0.117 -0.117 -0.1715 -0.1727 -0.1886 -0.1886...
 -0.1886];   % Fatigue exponent
A=-0.031632;  BC=0.003;  Q=110000;   SE=0.329;    % Creep constants
        %   A=-0.031632;  BC=0.003055;  Q=106000;   SE=0.329;    % Creep constants
%
% Failure Analysis postprocessor
    % Creep Analysis
    creep1=exp((1./(A + BC.*log(Sc1)))+(Q./(Tc1 + 459.67)) + CN*SE);
    creep3=exp((1./(A + BC.*log(Sc3)))+(Q./(Tc3+459.67))+CN*SE);
    % Fatigue Analysis
    fat3=0.5*(0.5*Sf3./interp1(TEMP,SFP,Tf3)./(1-0.5*Sf3./interp1(TEMP,SU,Tf3)))...
    .^(1./interp1(TEMP,SB,Tf3));
%
% Response Output Matrix
Y=[Tc1' Sc1' creep1' Tc3' Sc3' creep3' Tf3' Sf3' fat3'];
clear Sc1 Tc1 Sc3 Tc3 Sf3 Tf3 creep1 creep3 fat3 CN;
%
% Find variance and correlation coefficient matrices
covy=cov(Y);
%save RSEcovy.dat covy -ascii
corry=corrcoef(Y);
%save RSEcorry.dat corry -ascii

% ----------------------------------
% Empirical plots Generation
% ----------------------------------
%
% Empirical plots Points calculations
Failure(:,1)=sort(Y(:,3));
Failure(:,2)=sort(Y(:,6));
Failure(:,3)=sort(Y(:,9));
%
i=[1:x];
pFailure=i/x;
% Joint Prob CDF
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
k=1;
for j=[60:60:6000]
    C=0;
    for i=[1:x]
        if Failure(i,1)<j | Failure(i,2)<j | Failure(i,3)<j
        C=C+1;
        end
    end
    Pf(k)=C/x;
    k=k+1
end
life=[60:60:6000];
save jpfint.dat Pf -ascii
%
plot(Failure(:,1),pFailure,'-',Failure(:,2),pFailure,'-.',Failure(:,3),...
pFailure,'--',life,Pf,':')
xlabel('Time');
ylabel('Cumulative Probability');
axis([0 6000 0 0.02]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',0);
% Temperature response plots
[N,c1]=hist(Y(:,1),1000);
[N,c3]=hist(Y(:,4),1000);
[N,f3]=hist(Y(:,7),1000);
plot(c1,N,'-',c3,N,'-.',f3,N,':')
xlabel('Temperature (Fahrenheit)');
ylabel('Frequency');
axis([0 5E10 0 2e4]);
legend('Region 1 Temperature','Region 3 Temperature');
legend('Region 1 Temperature','Region 3 Temperature',0);
% Failure response plots
[N,c1]=hist(Y(:,3),1000);
[N,c3]=hist(Y(:,6),1000);
[N,f3]=hist(Y(:,9),1000);
plot(c1,N,'-',c3,N,'-.',f3,N,':')
xlabel('Life (hours)');
ylabel('Frequency');
axis([0 5E10 0 3e4]);
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)');
legend('Creep (Region 1)','Creep (Region 3)','Fatigue (Region 3)',0);
```

```
%
% ------------------------------------------------
% PROBABILITY ANALYSES                            ----
% ------------------------------------------------
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Independent Input)
% ------------------------------------------------------------------------------
%
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Independent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 10 yrs
fatlimit=5475;    % avg of 5.0 trips/cycles per day, 365 days per year, for 10 years
gulimit=0.8;
% Series (Ind) Probability calculation P(Y1>=maxy1)*P(Y2>=maxy2)
clear C
C=[0 0 0];
for j=[3 6 9];
    for i=[1:x];
        if j==1 | j==2;
            limit=creeplimit;
        else
            limit=fatlimit;
        end
        if Y(i,j)<=limit
        C(j/3)=C(j/3)+1;
        end
    end
end
Pfcs=1-(1-C(1)/x)*(1-C(2)/x)*(1-C(3)/x);        % Series event (ind) probability formula
                % Pfcs=0.0219   Pf=[0.0000    0.0141    0.0079]    10/14/03
%
% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Dependent Responses, Independent Input)-
% ------------------------------------------------------------------------------
%
%


% ------------------------------------------------------------------------------
% Hot Spot Failure Probability Calculation (Dependent Responses,Dependent Input)
% ------------------------------------------------------------------------------
%
% Define failure conditions of three responses (See Younghans et al. 200?)
creeplimit=5475;  % avg of 5 hrs per day, 365 days per year, for 10 yrs
fatlimit=5475;    % avg of 5.0 trips/cycles per day, 365 days per year, for 10 years
gulimit=0.8;
%
% Joint Probability calculation P(Y1>=maxy1 OR Y2>=maxy2)
C=0;
for i=[1:x];
        if Y(i,3)<=creeplimit | Y(i,6)<=creeplimit | Y(i,9)<=fatlimit
        C=C+1;
        end
end
%clear y
Pfcd=C/x        % Joint probability
                % Pfcd=0.0151   10/14/03
%
%
```

# REFERENCES


[1] "Nimonic alloy 80A," online technical report (july 2003), Special Metals, www.specialtymetals.com.

[2] "Navy/NASA engine program (NNEP89): A user's manual," August 1991.

[3] "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment," *SAE ARP 4761*, December 1996.

[4] "NASA CFM56 thermodynamic 1-D cycle model." NASA Glenn, September 2000.

[5] "Strategic plan for safety, security, and system efficiency," planning document, FAA, January 2001.

[6] Anderson, T. and Darling, D., "A test of goodness-of-fit," *Journal of the American Statistical Association*, vol. 49, pp. 765–769, 1954.

[7] Ang, A.-S. and Cornell, C., "Reliability bases of structural safety and design," *Journal of the Structural Division, ASCE*, vol. 100, no. ST9, pp. 1755–1769, 1974.

[8] Annis, C., "A strategy for improving 'Confidence' in a failure probability," *AIAA2003-1572, 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Confere*, 2003.

[9] Ayyub, B. and MCuen, R., *Numerical Methods for Engineers*. Upper Saddle River, NJ: Prentice Hall, 1996.

[10] Bandte, O., Mavris, D., and DeLaurentis, D., "Viable designs through a joint probabilistic estimation technique," *AIAA-99-5623, Proceedings, 18th Wind Energy Symposium, 37th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 11-14, 1999.

[11] Bandte, O., *A Probabilistic Multi-Criterion Decision Making Technique for Conceptual and Preliminary Aerospace Systems Design*. PhD dissertation, Georgia Institute of Technology, School of Aerospace Engineering, September 2000.

[12] Basquin, O., "The exponential law of endurance tests," *Proceedings of the American Society for Testing and Materials*, vol. 10, pp. 625–630, 1910.

[13] Bier, V., "On the state of the art: Risk communication to the public," *Reliability Engineering and Systems Safety*, vol. 71, pp. 139–150, 2001.

[14] Birnbaum, Z., "On the importance of different components in a multicomponent system," in *Multivariate Analysis* (Krishnaiah, P., ed.), (San Diego), pp. 581–592, Academic Press, 1969.

[15] Blanchard, B. and Fabrycky, W., *Systems Engineering and Analysis*. New York: Prentice-Hall, 3rd ed., 1998.

[16] Bourgund, U., Ouypornprasert, W., and Prenninger, P., "Advanced simulation methods for the estimation of system reliability," internal working report no. 19, Institut für Mechanik, Universität Innsbruck, Innsbruck, Austria, November 1986.

[17] Box, G., Hunter, W., and Hunter, J., *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. New York, NY: Wiley, 1978.

[18] Box, G. and Wilson, K., "On the experimental attainment of optimum conditions," *Journal of Royal Statistical Society*, vol. B13, pp. 1–38, 1951.

[19] Breitung, K., "Asypmtotic approximation for multinormal integrals," *Journal of Engineering Mechanics, ASCE*, vol. 110, no. 3, pp. 357–366, 1984.

[20] Burton, S., Tappeta, R., Kolonay, R., and Padmanabhan, D., "Turbine blade reliability-based optimization using a variable-complexity method," *AIAA-2002-1710, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 4th AIAA Non-Deterministic Approaches Forum*, Denver, CO, April 22–25, 2002.

[21] Cornell, C., "A probability-based structural code," *Journal of the American Concrete Institute*, vol. 66, pp. 147–163, 1969.

[22] Cowles, B., "High cycle fatigue in aircraft gas turbines-an industry perspective," *International Journal of Fracture*, vol. 80, pp. 147–163, 1996.

[23] Daleo, J.A., E.-K. and Woodford, D., "Application of stress relaxation testing in metallurgical life assessment evaluations of GTD111 alloy turbine buckets," *Journal of Engineering for Gas Turbines and Power*, vol. 121, pp. 129–137, January 1999.

[24] Daleo, J. and Wilson, J., "GTD111 alloy material study," *Journal of Engineering for Gas Turbines and Power*, vol. 120, pp. 375–382, January 1999.

[25] David, H. and Moeschberger, M., *The Theory of Competing Risks*. New York, NY: Macmillan Publishing, 1978.

[26] Deiter, G., *Engineering Design: A Materials and Processing Approach*. New York: McGraw-Hill, 2000.

[27] DeLaurentis, D. and Mavris, D., "Uncertainty modeling and management in multidisciplinary analysis and synthesis," *AIAA-2000-0422, 38th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 10-13 2000.

[28] Der Kiureghian, A., Lin, H.-Z., and Hwang, S.-J., "Second-order reliability approximations," *Journal of Engineering Mechanics, ASCE*, vol. 113, no. 8, pp. 1208–1225, 1987.

[29] Der Kiureghian, A. and Liu, P.-L., "Structural reliability under incomplete probability information," *Journal of Engineering Mechanics*, vol. 112, pp. 85–104, Junary 1985.

[30] Devore, J., *Probability and Statistics for Engineering and the Sciences*. New York, NY: Wadsworth, Inc., 1995.

[31] Ditlevson, O., "Narrow reliability bounds for structural systems," *Journal of Structural Mechanics*, vol. 7, no. 4, pp. 453–472, 1979.

[32] Dowling, N., "Fatigue prediction for complicated stress-strain histories," *Journal of Materials*, vol. 7, pp. 71–87, 1972.

[33] Dowling, N., *Mechanical Behavior of Materials*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1993.

[34] Dowling, N., *ASM Handbook: Vol. 19 Fatigue and Fracture*, ch. Estimating Fatigue Life. Materials Park, OH: ASM International, 1996.

[35] Dubi, A., *Monte Carlo Applications in Systems Engineering*. New York, NY: John Wiley & Sons, 2000.

[36] Fiessler, B., Newmann, H., and Rackwitz, R., "Quadratic limit states in structural reliability," *Journal of Engineering Mechanics, ASCE*, vol. 105, no. EM4, pp. 661–676, 1979.

[37] Fox, E. and Safie, F., "Statistical characterization of life drivers for a probabilistic design analysis," *AIAA/SAE/ASME/ASEE 28th Joint Propulsion Conference and Exhibit*, pp. 911–919, Nashville, TN, July 6-8 1992.

[38] Garvey, P., *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*. New York, NY: Marcel Dekker, 1999.

[39] Ghanem, R.-G. and Spanos, P.-D., *Stochastic Finite Elements-a spectral approach*. Berlin: Springer, 1994.

[40] Ghiocel, D., "Stochastic field models for aircraft jet engines," *Journal of Aerospace Engineering, ASCE*, vol. 14, pp. 127–139, October 2001.

[41] Gikhman, I. and Skorokhod, A., *Introduction to the Theory of Random Processes*. Mineola, NY: Dover Publications, 1996.

[42] Gollwitzer, S. and Rackwitz, R., "An efficient numerical solution to the mutlinormal integral," *Probabilistic Engineering Mechanics*, vol. 3, no. 2, pp. 98–101, 1988.

[43] Golomski, W., "Reliability and social policy," *IEEE Transactions on Reliability*, vol. 50, no. 2, pp. 131–150, 2001.

[44] Gore, A., "Final report to president clinton: White house commission on aviation safety and security." 1997.

[45] Haldar, A. and Mahadevan, S., *Probability, Reliability, and Statistical Methods in Engineering Design*. New York, NY: John Wiley and Sons, 2000.

[46] Hasofer, A. and Lind, N., "Exact and invariant second moment code format," *Journal of the Engineering Mechanics Division, ASCE*, vol. 100, no. EM1, pp. 771–791, 1985.

[47] Hotelling, H., "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, December 1936.

[48] Høyland, A. and Rausand, M., *System Reliability Theory: Models and Statistical Methods*. New York, NY: John Wiley & Sons, 1994.

[49] Jarque, C. and Bera, A. K., "A test for normality of observations and regression," *International Statistical Review*, vol. 55, pp. 163–172.

[50] Jeschke, P., J.-J. S. R. and Riegler, C., "Preliminary gas turbine design using the multidisciplinary design system MOPEDS," *GT-2002-30496, Proceedings of ASME Turbo Expo*, Amsterdam, The Netherlands, June 3-6 2002.

[51] Koch, P., Mavris, D., and Mistree, F., "Multi-level, partitioned response surfaces for modeling complex systems," *AIAA-98-4958, 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998.

[52] Lawless, J., *Statistical Models and Methods for Lifetime Data*. New York, NY: Wiley, 1982.

[53] Leemis, L., *Reliability: Probabilistic Models and Statistical Methods*. Upper Sadle River, NJ: Prentice Hall, 1995.

[54] Li, C.-C. and Der Kiureghian, A., "Optimal discretization of random fields," *Journal of Engineering Mechanics, ASCE*, vol. 119, no. 6, pp. 1136–1154, 1993.

[55] Liu, P.-L. and Der Kiureghian, A., "Multivariate distribution models with prescribed marginals and covariances," *Probabilistic Engineering Mechanics*, vol. 1, pp. 105–112, January 1986.

[56] Liu, W.-K., Belytschko, T., and Mani, A., "Probabilistic finite elements for non-linear structural dynamics," *Comput. Meth. Appl. Mech. Engng*, vol. 56, pp. 61–86, 1986.

[57] Liu, W.-K., Belytschko, T., and Mani, A., "Random field finite elements," *International Journal of Numerical Methods in Engineering*, vol. 23, no. 10, pp. 1831–1845, 1986.

[58] Liu, Z., W.-J. and Mavris, D., "Probabilistic oxidation life assessment of a hot gas path component," *6th AIAA Non-Deterministic Approaches Forum*, Palm Springs, CA, April 19-22 2004.

[59] Liu, Z., *Methodology for Probabilistic Remaining Life Assessment of a Gas Turbine Airfoil*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, 2002.

[60] Liu, Z., Mavris, D., and Volovoi, V., "Creep life prediction of gas turbine components under varying operating conditions," *JPGC-2001-PWR-19163 , ASME International Joint Power Generation Conference*, New Orleans, LA, June 4–7 2001.

[61] Madeson, H., Krenk, S., and Lind, N., *Methods of Structural Safety*. Englewood Cliffs, NJ: Prentice Hall, 1986.

[62] Mahadevan, S., *Stochastic Finite Element-Based Structural Reliability Analysis and Optimization*. PhD thesis, Georgia Institute of Technology, Atlanta, GA, July 1988.

[63] Mahadevan, S., Dey, A., Tryon, R., Wang, Y., and Rousseau, C., "Reliability analysis of rotorcraft composite structures," *Journal of Aerospace Engineering, ASCE*, vol. 14, pp. 140–146, October 2001.

[64] Mavris, D. and Roth, B., "A methodology for robust design of impingement cooled hsct combustion liners," *AIAA-97-0288, 35th Annual Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 6–9 1997.

[65] Metals, S., "Technical bulletin: Nimonic alloy 105," tech. rep., Special Metals Corporation, www.specialmetals.com, September 2003.

[66] Moon, D., Simon, R., and Favor, R., "Elevated temperature properties of selected superalloys," *ASTM Data Series DS 7-S1*.

[67] Morgenstern, D., "Einache beispele zweidimensionaler verteilungen," *Miteilingsblatt fur Mathematische Statistik*, vol. 8, pp. 234–235, 1956.

[68] Morrow, J., *Fatigue Design Handbook*, vol. 4, Sec. 3.2, ch. Advances in Engineering, pp. 22–29. Society of Automotive Engineers, 1968.

[69] Nataf, A., "Determination of des distribution dont les marges sont donnees," *Comptes Rendus de l'Academi des Sciences*, vol. 225, pp. 42–43, 1962.

[70] Newell, J.F., R.-K. and Ho, H., "Probabilistic structural analysis of space propulsion system turbine blades," *AIAA-89-1372-CP, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Mobile, AL, 1989.

[71] O'Conner, P., "Commentary: Reliablity-past, present, and future," *IEEE Transactions on Reliability*, vol. 49, no. 4, pp. 335–341, 2000.

[72] Orr, R.L., S.-O. and Dorn, J., "Correlations of creep rupture data for metals at elevated temperatures," *Trans. ASM*, vol. 46, pp. 113–124, 1954.

[73] Pascual, F. and Meeker, W., "Estimating fatigue curves with the random fatigue limit model," *Technometrics*, vol. 41, pp. 277–302, November 1999.

[74] Rackwitz, R. and Fiessler, B., "Note on discrete safety checking when using non-normal stochastic models for basic variables," loads project working session, MIT, Cambridge, MA, 1978.

[75] Reed, R.P., S.-J. and Christ, B., "The economic effects of fracture in the united states: Part 1," special pub. no. 647-1, U.S. Dept. of commerce, National Bureau of Standards, Washington, DC, 1983.

[76] Rencher, A., *Methods of Multivariate Analysis*. New York: John Wiley & Sons, 1995.

[77] Rosenblatt, M., "Remarks on a multivariate transformation," *Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 470–472, 1952.

[78] Rustenburg, J., S.-D. and Tipps, D., "Statistical loads data for Boeing 737-400 aircraft in commercial operations," DOT/FAA/AR-98/28.

[79] Shooman, M., *Probabilistic Reliability, an Engineering Approach*. New York, NY: McGraw-Hill, 1968.

[80] Singpurwalla, N., "Survival in dynamic environments," *Statistical Science*, vol. 10, pp. 86–103, February 1995.

[81] Southwest Research Institute, *FPI Software and User's Manual (version 6.20)*, 1995.

[82] Stephens, M., "EDF statistics for goodness of fit and some comparison," *Journal of American Statistical Association*, vol. 69, pp. 730–737, 1974.

[83] Stephens, M., "Asymptotic results for goodness-of-fit statistics with unknown parameters," *Annal of Statistics*, vol. 4, pp. 357–369, 1976.

[84] Stephens, M., "Goodness of fit for the extreme value distribution," *Biometrika*, vol. 64, pp. 583–588, 1977.

[85] Stephens, M., "Goodness of fit with special reference to tests for exponentiality," technical report no. 262, Stanford University, 1977.

[86] Stephens, M., "Tests of fit for the logistic distribution based on empirical distribution function," *Biometrika*, vol. 66, pp. 591–595, 1979.

[87] Sues, R.H., C.-M. P. S. and Wu, J. Y.-T., "Reliability-based optimization considering manufacturing and operational uncertainties," *Journal of Aerospace Engineering*, vol. 14, pp. 166–174, October 2001.

[88] Suresh, S., *Fatigue of Materials*. Cambridge, United Kingdom: Cambridge University Press, 1998.

[89] Tappeta, R., Kolonay, R., and Burton, S., "Application of approximate optimization to turbine blade design in a network-centric environment," *AIAA-2002-1588, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 4th AIAA Non-Deterministic Approaches Forum*, Denver, CO, April 22-25 2002.

[90] Tinga, T., Visser, W., de Wolf, W., and Broomhead, M., "Integrated lifing analysis tool for gas turbine components," *NLR-TP-2000-049, ASME Turbo Expo*, Munich, Germany, May 8–11 2000.

[91] Tong, Y., *The Multivariate Normal Distribution*. New York, NY: Springer Verlag, 1990.

[92] Toucher, K., *The Art of Simulation*. London: English Universities Press, 1963.

[93] Wallace, J. and Mavris, D., "Propulsion system reliability prediction and optimization using a multi-physics environment," *AIAA 2002-5561, 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA, September 4–6 2002.

[94] Wallace, J. and Mavris, D., "Creep life uncertainty assessment of a gas turbine airfoil," *AIAA2003-1484, 5th AIAA Non-Deterministic Approaches Forum*, Norfolk, VA, April 7-10, 2003.

[95] Wallace, J.M., L.-Z. W. R. V. V. and Mavris, D., "Annual progress report: Probabilistic life assessments of hot gas path components," internal technical report, General Electric Power Systems, Greenville, SC, May 2002.

[96] Wallace, J.M., W.-S. and Mavris, D., "Robust design of a creep limited gas turbine component using the taguchi approach," *GT2003-38546, ASME Turbo Exposition*, June 16-19 2003.

[97] Weibull, W., "A statistical distribution function of wide applicability," *Journal of Applied Mechanics*, vol. 18, pp. 293–297, 1951.

[98] Wöhler, A., "Versuche über die festigkeit der eisenbahnwagenachsen," *Zetschrift für Bauwesen*, vol. 10, 1860.

[99] Wu, Y.-T., M. H. and Cruse, T., "Advanced probabilistic structural analysis method for implicit performance functions," *AIAA Journal*, vol. 28, pp. 1663–1669, September 1990.

[100] Wu, Y.-T., "An adaptive importance sampling method for structural system reliability analysis," *Reliability Technology*, vol. 28, pp. 217–231, 1992.

[101] Wu, Y.-T., "FAST probability integration: Concepts and applications." Presentation at Georgia Tech Aerospace Systems Design Laboratory, Atlanta, GA, June 1999.

# VITA

Jon Michael Wallace was born in Ft. Campbell, KY on September 18, 1975. He spent most of his pre-college years in Fayetteville, NC and graduated from the Terry Sanford Senior High School in 1993. The author entered the Georgia Institute of Technology in 1993 where he received the Bachelor of Science Mechanical Engineering in June of 1999. During his Bachelors program he managed to complete eleven successful co-operative engineering positions with companies including Delta Air Lines, Black & Decker, and General Electric Aircraft Engines. In the fall of 1999 he enrolled in the graduate program also at the Georgia Institute of Technology. He received the Master of Science in Mechanical Engineering in August of 2000, and a Master of Science in Aerospace Engineering in December of 2002.